# Explicit-in-Time Variational Formulations for Goal-Oriented Adaptivity

*Judit Muñoz-Matute*

Supervised by *David Pardo* and *Elisabete Alberdi*

July 2019

eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

basque center for applied mathematics

# Explicit-in-Time Variational Formulations for Goal-Oriented Adaptivity

*Judit Muñoz-Matute*

Supervised by *David Pardo* and *Elisabete Alberdi*

July 2019

This dissertation has been possible with the support of the University of the Basque Country (UPV/EHU) grant No. PIF15/346; the Basque Government through the BERC 2014-2017 program and the Consolidated Research Group Grant IT649-13 on "Mathematical Modeling, Simulation, and Industrial Applications (M2SI)"; the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2016-76329-R (AEI/FEDER, EU), and MTM2016-81697-ERC/AEI; the BCAM "Severo Ochoa" accreditations of excellence SEV-2017-0718; the National Science Centre, Poland grant no. 2017/26/M/ST1/00281; the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreements No 644602 (GEAGAM) and 777778 (MATROCKS).

# Acknowledgements

It has been a long journey since I started this adventure four years ago. First, I want to thank from my heart to my supervisor and now friend Prof. David Pardo. His guidance along these years has been simply perfect. Among other virtues, I would like to highlight his enthusiasm in research, his intuition, and professionalism. Also, his ability to keep a good balance: both at work, because he is capable to be the head of a strong research group that at the same time is like a family; and in life, because as he says: *even something that seems simple can go wrong and nothing that seems impossible is unreachable.*

I would also like to thank my supervisor Prof. Elisabete Alberdi for trusting me from the very beginning and counting on me for every project, always thinking about my benefit. Her dedication and hard work have been admirable. Thanks for teaching me how the University works and for giving me the opportunity to teach as well.

I am deeply grateful to Prof. Victor Calo for hosting me in Perth for eight months, for sharing with me his boundless knowledge and his passion for science. But mostly, I want to thank him and his family for making me feel at home, while being literally, in the other side of the world. I also want to thank Prof. Maciej Paszyński for inviting me to Krakow twice to collaborate with his group and for all the laughs we have shared in every destination we have coincided. I think I will live longer now.

I express my gratitude to Prof. Ignacio Muga and Prof. Kris Van der Zee for hosting me in Valparaíso and Nottingham, respectively. I learnt from them to face (and sometimes survive) challenging questions, that being an expert in functional analysis is compatible with loving hard rock music, and that even in a bar grabbing a beer it is possible to have fruitful discussions about mathematics.

I want to offer a special thank to Prof. Leszek Demkowicz for hosting me in Austin during the last months of my thesis and also Prof. Jay Gopalakrishnan for all the interesting discussions we had in Bilbao and New York. I could not be more grateful to have had the opportunity to discuss the topics of this dissertation with them.

I wish to thank all my colleagues from the MATHMODE group, BCAM and Victor's group in Perth for their friendship and all the great moments. Especially, Vincent Darrigrand, Daniel García, and Mostafa Shahriari for answering every annoying question both about science and bureaucracy and for all the time we

have spent together here in Bilbao and around the world.

Finally, I want to thank my family and friends for their support. In particular, I wish to offer my eternal gratitude to my brothers and the three people that have closely followed and sometimes suffered my evolution as a researcher since I started studying mathematics ten years ago. First, to my parents for their unconditional support and love, and the values they have taught to me that have made possible who I am today. Last but not least, to my partner Jon for being the serenity I needed in difficult times and for always trusting so strongly in me (and more than myself) in all that I am able to achieve. I love you all.

# Abstract

Goal-Oriented Adaptivity (GOA) is a powerful tool to accurately approximate physically relevant features of the solution of Partial Differential Equations (PDEs). It delivers optimal grids to solve challenging engineering problems. In time dependent problems, GOA requires to represent the error in the Quantity of Interest (QoI) as an integral over the whole space-time domain in order to reduce it via adaptive refinements. A full space-time variational formulation of the problem allows the aforementioned error representation. Thus, variational space-time formulations for PDEs have been of great interest in the last decades, among other things, because they allow to develop mesh-adaptive algorithms. Since it is known that implicit time marching schemes have variational structure, they are often employed for GOA in time-domain problems. When coming to explicit-in-time methods, these were introduced for Ordinary Differential Equations (ODEs) employing specific inexact quadrature rules.

In this dissertation, we prove that the explicit Runge-Kutta (RK) methods can be expressed as discontinuous-in-time Petrov-Galerkin (dPG) methods for the linear advection-diffusion equation. We systematically build trial and test functions that, after exact integration in time, lead to one, two, and general stage explicit RK methods. This approach enables us to reproduce the existing time-domain goal-oriented adaptive algorithms using explicit methods in time. Here, we employ the lowest order dPG formulation that we propose to recover the Forward Euler method and we derive an appropriate error representation. Then, we propose an explicit-in-time goal-oriented adaptive algorithm that performs local refinements in space. In terms of time domain adaptivity, we impose the Courant-Friedrichs-Lewy (CFL) condition to ensure the stability of the method. We provide some numerical results in one-dimensional (1D)+time for the diffusion and advection-diffusion equations to show the performance of the proposed algorithm.

On the other hand, time-domain adaptive algorithms involve solving a dual problem that runs backwards in time. This process is, in general, computationally expensive in terms of memory storage. In this work, we define a pseudo-dual problem that runs forwards in time. We also describe a forward-in-time adaptive algorithm that works for some specific problems. Although it is not possible to define a general dual problem running forwards in time that provides information about future states, we provide numerical evidence via one-dimensional

problems in space to illustrate the efficiency of our algorithm as well as its limitations. As a complementary method, we propose a hybrid algorithm that employs the classical backward-in-time dual problem once and then performs the adaptive process forwards in time. We also generalize a novel error representation for goal-oriented adaptivity using (unconventional) pseudo-dual problems in the context of frequency-domain wave-propagation problems to the time-dependent wave equation. We show via 1D+time numerical results that the upper bounds for the new error representation are sharper than the classical ones. Therefore, this new error representation can be used to design more efficient goal-oriented adaptive methodologies.

Finally, as classical Galerkin methods may lead to instabilities in advection-dominated-diffusion problems and therefore, inappropriate refinements, we propose a novel stabilized discretization method, which we call Isogeometric Residual Minimization (iGRM) with direction splitting. This method combines the benefits resulting from Isogeometric Analysis (IGA), residual minimization, and Alternating Direction Implicit (ADI) methods. We employ second order ADI time integrator schemes, B-spline basis functions in space and, at each time step, we solve a stabilized mixed method based on residual minimization. We show that the resulting system of linear equations has a Kronecker product structure, which results in a linear computational cost of the direct solver, even using implicit time integration schemes together with the stabilized mixed formulation. We test our method in 2D and 3D+time advection-diffusion problems. The derivation of a time-domain goal-oriented strategy based on iGRM will be considered in future works.

# Resumen

Las Ecuaciones en Derivadas Parciales (EDPs) están muy presentes en nuestra vida diaria ya que modelan múltiples procesos físicos. Podemos encontrar distintas aplicaciones de EDPs en modelos de turbulencia, pronóstico del tiempo, simulación de terremotos, diseño de antenas y satélites o modelos de crecimiento de tumores entre otros muchos. Normalmente, las soluciones analíticas de estos modelos se desconocen, por lo que es esencial desarrollar métodos numéricos de alta precisión para realizar simulaciones de calidad. Hoy en día, el Método de Elementos Finitos (MEF) [80, 133] es una técnica comúnmente utilizada para aproximar soluciones de EDPs. La flexibilidad de la descripción geométrica que proporciona el MEF permite que este método modele una amplia variedad de problemas de ingeniería. Los algoritmos adaptativos en el MEF [44, 51] son herramientas esenciales para producir mallas óptimas y así obtener soluciones precisas minimizando el costo computacional. En estos tipos de algoritmos, el tamaño del elemento y/o el orden de aproximación varían localmente. Los procesos adaptativos utilizan indicadores locales del error que se diseñaron inicialmente para reducir el error global en la norma de la energía.

Sin embargo, en muchos problemas de ingeniería, es esencial aproximar con precisión algunas características físicas relevantes de la solución. Dichas características se denominan cantidades de interés y generalmente se representan mediante funcionales de la solución. De esta necesidad, surgió la idea de "adaptatividad orientada a un objetivo" [15, 110, 120]. El objetivo de estos algoritmos adaptativos es reducir el error en una cierta cantidad de interés. Para ello, se define un problema dual cuya fuente es un funcional de la solución. Después, se representa el error en la cantidad de interés como una integral sobre todo el dominio utilizando los errores del problema directo y el dual. Por último, esta representación del error se acota superiormente por la suma de las contribuciones locales de cada elemento, las cuales se utilizan para dirigir el proceso adaptativo. Recientemente, en [40, 41] los autores han desarrollado una nueva representación del error para problemas de propagación de ondas en el dominio de la frecuencia. Han obtenido cotas superiores más finas que las clásicas para la ecuación de Helmholtz en múltiples dimensiones. Su método se basa en emplear una forma bilineal alternativa que exhibe mejores propiedades que la original. Utilizando este nuevo enfoque, se pueden mejorar los procesos adaptativos existentes para los problemas de propagación de ondas.

*Resumen*

En esta tesis doctoral, nos centramos principalmente en algoritmos adaptativos orientados a un objetivo para EDPs que dependen del tiempo. Para la discretización de problemas temporales es muy común emplear el llamado método de líneas [126]. En este método, las variables espacio y tiempo se discretizan de forma independiente. Primero, a partir de una formulación variacional en espacio, la variable espacial se discretiza empleando el MEF. Después, se obtiene un sistema de Ecuaciones Diferenciales Ordinarias (EDOs) que se puede discretizar utilizando métodos clásicos de diferencias finitas [26, 76]. En el método de líneas, la adaptatividad del mallado se realiza independientemente en espacio (localmente) y en tiempo (local en tiempo pero global en espacio). La idea alternativa de utilizar métodos variacionales espacio-tiempo se estableció a finales de los ochenta y principios de los noventa [7, 67]. Existen varios algoritmos adaptativos basados en este método, por ejemplo, estrategias que utilizan mallas espacio-temporales no estructuradas [70]. En otros algoritmos, los autores asumen un producto tensorial espacio-tiempo de las funciones de base. Luego, empleando funciones de base discontinuas en tiempo, el MEF en espacio-tiempo se puede reinterpretar como un método de líneas. Aquí se pueden adaptar tanto los órdenes de aproximación como el tamaño de la malla en espacio y tiempo [130, 147].

De manera similar al MEF en espacio-tiempo, para realizar adaptatividad orientada a un objetivo para EDPs que dependen del tiempo, también se requiere una formulación variacional espacio-tiempo del problema. Aquí, representamos el error en la cantidad de interés como una integral en todo el dominio espacio-temporal, empleando los errores del problema directo y dual. Una formulación variacional espacio-tiempo permite dicha representación. Basándose en esta formulación, se han desarrollado diferentes estrategias adaptativas orientadas a un objetivo para EDPs en el dominio del tiempo [9, 54, 136].

Existen varios problemas relacionados con la adaptatividad orientada a un objetivo en el dominio del tiempo que abordamos en esta tesis doctoral. Uno de ellos está relacionado con el problema dual que también está gobernado por una EDP pero, al contrario que el problema directo, evoluciona hacia atrás en el tiempo. Por lo tanto, los proceso adaptativos orientados a un objetivo requieren resolver dos problemas que evolucionan en direcciones opuestas en tiempo. Este hecho implica que es necesario almacenar toda la información proveniente de ambos problemas para estimar las contribuciones locales del error y realizar la adaptatividad.

Por otro lado, para discretizar los problemas primal y dual utilizando el método de líneas, es esencial conocer la equivalencia con su correspondiente MEF en espacio-tiempo. Sin embargo, las formulaciones variacionales espacio-temporales más utilizadas para resolver EDPs son equivalentes a métodos implícitos en tiempo cuando se reinterpretan como un método de líneas. Por esa razón, la

mayoría de los procesos adaptativos orientados a un objetivo emplean métodos implícitos en tiempo. Nuestro objetivo es diseñar algoritmos adaptativos orientados a un objetivo utilizando métodos explícitos en tiempo, ya que en muchos casos pueden ser computacionalmente más baratos que los implícitos. Para ello, necesitamos derivar una formulación variacional para métodos explícitos en tiempo. Existen algunos trabajos que describen formulaciones variacionales de métodos explícitos para EDOs seleccionando algunas reglas de cuadratura específicas [38, 150]. Sin embargo, con esta metodología, es posible recuperar algunos métodos explícitos de tipo Runge-Kutta (RK) pero únicamente caso por caso y no de forma constructiva.

Finalmente, también abordamos el hecho de que si el método numérico empleado para la discretización es inestable, los algoritmos adaptativos introducirían refinamientos en lugares innecesarios. Para evitar este problema, es preferible emplear métodos de discretización estables. Existe una literatura extensa sobre métodos estabilizados para EDPs, especialmente para problemas de advección-difusión [25, 79, 82]. Entre otros, el método Petrov-Galerkin discontínuo (DPG) con funciones test óptimas, que es una clase de método de minimización residual [45, 46]. La idea clave del método DPG es construir una aproximación de las funciones de test óptimas en tiempo real, elemento por elemento y garantizando automáticamente la estabilidad numérica del método. Recientemente, se ha introducido el dual del método DPG, llamado DPG* [48, 92]. En este último, al igual que en DPG, se resuelve un problema de punto de silla pero con una cantidad de interés en el lado derecho. Esta nueva metodología permite realizar refinamientos adaptativos orientados a un objetivo utilizando métodos DPG.

Junto con los métodos de minimización residual como DPG, consideraremos la combinación de otros dos métodos de discretización: los métodos de análisis isogeométrico (IGA por sus siglas en inglés) y los métodos implícitos de dirección alterna (ADI por sus siglas en inglés). IGA es un método moderno para realizar simulaciones de elementos finitos utilizando B-splines y NURBS [39, 81], lo que resulta en una aproximación de la solución más suave y de mayor orden. Los métodos ADI [39, 81] son populares para realizar simulaciones de diferencias finitas en mallas regulares. Este método introduce pasos de tiempo intermedios, y el operador diferencial espacial se divide en las componentes $x$, $y$ y $z$. Como resultado, en el lado izquierdo, solo tenemos derivadas en una dirección, mientras que el resto del operador está en el lado derecho.

En esta tesis doctoral, abordamos los problemas mencionados anteriormente relacionados con la adaptatividad orientada a un objetivo en el dominio del tiempo, resumiendo los principales resultados de nuestras publicaciones [97, 98, 105, 106, 107, 108].

En [108] proponemos un método constructivo para derivar formulaciones varia-

cionales de esquemas explícitos de tipo RK de cualquier número de etapas para EDPs parabólicas lineales. Primero, derivamos una formulación Petrov-Galerkin discontinua en tiempo donde definimos los saltos de la solución utilizando una aproximación "downwind" en cada interfaz temporal. En nuestra formulación permitimos diferentes discretizaciones espaciales por intervalo temporal. Para un número general de etapas, definimos familias de polinomios a trozos y los sustituimos en la formulación variacional. Al tratar los coeficientes de los polinomios como incógnitas, obtenemos integrales de los productos de las funciones de base. Luego, establecemos algunas condiciones que deben cumplir estas integrales en tiempo. Primero, determinamos las condiciones de ortogonalidad necesarias para obtener un método explícito. También definimos las condiciones de no-ortogonalidad identificando las integrales restantes con las entradas de la tabla de Butcher que definen los métodos RK. Finalmente, integrando analíticamente en tiempo, obtenemos un sistema de ecuaciones no lineales. Al resolver este sistema, producimos los coeficientes de las funciones de base para cualquier método RK. Definimos el espacio de la solución y el espacio test correspondientes como los espacios generados por estas funciones. De esta forma, al caracterizar los espacios, podemos representar el error en la cantidad de interés de una forma natural de la misma manera que con los métodos implícitos.

En [106], siguiendo las ideas presentadas en [108], derivamos una representación espacio-temporal del error en la cantidad de interés y desarrollamos un proceso adaptativo orientado a un objetivo en espacio empleando métodos explícitos en tiempo. Permitimos mallas dinámicas en espacio, es decir, consideramos diferentes discretizaciones espaciales por intervalo de tiempo. Después, seleccionando funciones de base constantes a trozos obtenemos el método de Euler explícito en tiempo. Para discretizar el problema dual, intercambiamos los espacios de la solución y las funciones test con respecto al problema directo y también obtenemos el método Euler explícito pero hacia atrás en tiempo. Como estamos utilizando métodos explícitos en tiempo, la condición Courant-Friedrichs-Lewy (CFL) [132] debe imponerse para garantizar la estabilidad de la discretización. Nuestro algoritmo adaptativo se centra en realizar refinamientos óptimos en espacio seguido de una condición de refinamiento simple en tiempo con el objetivo de cumplir la condición CFL tras los refinamientos espaciales.

En [107], siguiendo las ideas presentadas en [40, 41], definimos un problema pseudo-dual que, al igual que el problema directo, avanza hacia delante en tiempo. Después, definimos un proceso adaptativo orientado a un objetivo también hacia delante en tiempo. En el algoritmo propuesto, resolvemos los problemas primal y dual en un paso de tiempo fijo para después adaptar el mallado espacial correspondiente antes de pasar al siguiente paso de tiempo. De esta forma, podemos realizar la adaptatividad a medida que calculamos la solución, por lo que

no necesitamos almacenarla para todos los pasos de tiempo en cada iteración. Sin embargo, este algoritmo solo funciona en algunas configuraciones específicas: por ejemplo, en problemas de difusión donde la cantidad de interés se localiza en un área del dominio espacial y tiene soporte en todo el intervalo temporal. Para problemas de advección-difusión y para problemas en los que la cantidad de interés esté concentrada en un intervalo de tiempo reducido, el problema pseudo-dual que definimos es inapropiado. Como alternativa para estos problemas de advección-difusión, proponemos un algoritmo híbrido que resuelve el problema dual clásico una vez hacia atrás en tiempo. Luego, establecemos la condición inicial del problema pseudo-dual como la condición inicial del problema dual clásico. Finalmente, el proceso adaptativo se realiza hacia delante en el tiempo. En ambos algoritmos, solo necesitamos almacenar las soluciones de los problemas primal y dual en un único paso de tiempo para realizar la adaptatividad.

En [105], extendemos la nueva representacion del error desarrollada en [40, 41] para la ecuación de Helmholtz a la ecuación de ondas en el dominio del tiempo. Seleccionamos una forma bilineal alternativa modificando en la clásica los términos que provienen de la segunda derivada temporal. Al hacerlo, creamos un problema pseudo-dual obteniendo otro problema de propagación de ondas con mejores propiedades de estabilidad. Comparamos las cotas superiores clásicas del error en la cantidad de interés con las nuevas, observando que los nuevos límites son más cercanos al error para la ecuación de ondas en una dimensión espacial. Por lo tanto, el método resultante proporciona un mejor criterio para guiar el proceso adaptativo.

Finalmente, en [97, 98], proponemos un método que reúne los beneficios de los métodos ADI, minimización residual e IGA. Nuestro objetivo es obtener de manera eficiente soluciones estables y precisas de problemas que dependen del tiempo. Primero, aplicamos varios esquemas de integración de tipo ADI de segundo orden. Después, estabilizamos el esquema resultante aplicando un método de minimización residual en cada paso de tiempo. Finalmente, realizamos la discretización espacial utilizando funciones de base que son producto tensorial de B-splines. De forma similar a DPG, resolvemos un problema de punto de silla en cada paso de tiempo. En nuestro método, conservamos la estructura de producto Kronecker en todo el sistema y de esta forma, podemos factorizar la matriz con un solucionador de costo computacional lineal. Mostramos a través de resultados numéricos en 2D y 3D+tiempo que este método corrige los problemas de estabilidad que surgen de los métodos clásicos de Galerkin para la ecuación de advección-difusión en el dominio del tiempo. Este método es el primer paso hacia la construcción de una estrategia adaptativa orientada a un objetivo (de manera similar al método DPG*) para las EDPs que dependen del tiempo empleando métodos estabilizados que consideraremos en el futuro.

# Contents

# Acronyms

**GOA** Goal-Oriented Adaptivity

**PDE** Partial Differential Equation

**ODE** Ordinary Differential Equation

**FEM** Finite Element Method

**SEM** Spectral Element Method

**IGA** Isogeometric Analysis

**dG** discontinuous-in-time Galerkin

**cPG** continuous-in-time Petrov-Galerkin

**dPG** discontinuous-in-time Petrov-Galerkin

**DoF** Degrees of Freedom

**QoI** Quantity of Interest

**MoL** Method of Lines

**RK** Runge-Kutta

**CFL** Courant-Friedrichs-Lewy

**ADI** Alternating Direction Implicit

**IGA** Isogeometric Analysis

**DPG** Discontinuous Petrov-Galerkin

**iGRM** Isogeometric Residual Minimization

# List of Figures

# List of Tables

# 1. Introduction

## 1.1. Motivation and literature review

Partial Differential Equations (PDEs) are present in our daily lives as they model multiple physical processes. We can find applications of these models in different areas such as aeronautics (turbulence modeling), meteorology (weather forecast), seismology (earthquake simulations), telecommunications (design of antennas and satellites) or medicine (tumor growth modeling). In most cases, analytical solutions of the above problems are either unavailable or intractable, so it is essential to develop high-precision numerical methods to perform accurate simulations. Nowadays, the Finite Element Method (FEM) [60, 80, 91, 95, 133] is a commonly used technique to approximate solutions of PDEs. The flexibility of the geometric description that FEM provides, allows the methodology to model a wide variety of engineering problems. Adaptive algorithms in FEM [44, 51, 53, 122] are essential tools to produce optimal grids in order to obtain accurate solutions while minimizing the computational cost. In these types of algorithms, the element size and/or the order of approximation vary locally. The adaptive processes use local indicators of the error that were initially designed to reduce the global error in the energy norm.

However, in many engineering problems, it is also important to accurately approximate some physical features of the solution. A relevant feature is referred to as Quantity of Interest (QoI) and they are usually represented by output functionals computed from the solution. From this need, the idea of "Goal-Oriented Adaptivity (GOA)" emerged. The seminal works developed by Becker and Rannacher [15, 16, 17] and the subsequent contributions of Oden and Prudhomme [110, 111, 120, 121] have been of great importance in this area. Goal-oriented adaptive strategies have been applied to a wide variety of problems [4, 31, 124] including electromagnetics [112, 113, 114, 115], fluid structure iterations [65, 137], free boundary problems [138, 139] and Isogeometric Analysis (IGA) [140, 94]. The objective of these adaptive algorithms is to reduce the error in the QoI. For that, a dual problem is defined where the output functional becomes the source. Then, an error in the QoI is represented as an integral over the whole domain in terms of the errors of the direct and dual problems. Finally, this error representation is bounded above by the sum of local element contributions that drive the goal-

oriented adaptive process [3]. Recently, Darrigrand et. al. [40, 41] developed a new error representation within the goal-oriented approach for frequency-domain problems. They obtained sharper upper bounds than the classical ones for the multidimensional Helmholtz equation. Their method is based on employing an alternative bilinear form that exhibits better properties than the original one. They represent the error in the QoI using both this alternative form and the original one. Using this new approach, the existing adaptive processes for wave propagation problems can be improved. In this dissertation, we focus mostly on goal-oriented adaptive algorithms for time-dependent PDEs and, among other things, we will extend the ideas from Darrigrand et. al. to the wave equation in time domain.

For the discretization of time-dependent problems, it is very common to employ the so-called Method of Lines (MoL) [80, 126]. In this method, the space and time variables are discretized independently. First, starting from a variational formulation in space [24, 96, 125], the spatial variable is discretized employing the FEM. Then, a first or a second order system of Ordinary Differential Equations (ODEs) is obtained for parabolic and hyperbolic problems, respectively. Traditional numerical methods such as Runge-Kutta (RK) methods or Backward Differentiation Formulae (BDF) [26, 76, 146] can be used for the resolution of first order ODEs. Second order ODE systems can be reduced to a first order ODE system and solved by the aforementioned numerical methods or directly using methods like the Hilber-Hughes-Taylor-$\alpha$ (HHT-$\alpha$) method [77, 148]. The generalized-$\alpha$ methods are a robust family of time integration techniques with dissipative control on the highest resolved frequencies for first and second order discretizations [30, 35, 88]. In the MoL, the mesh adaptivity is performed independently [18, 34, 37, 109] in space (locally) and time (local in time but globally in space) employing *a posteriori* error estimates (see Figure 1.1).

The alternative idea of using variational space-time methods was well established in the late eighties and early nineties [7, 59, 67, 90]. Hughes and Hulbert [84, 85] proposed a stabilized space-time FEM for hyperbolic problems. They showed that the oscillations present in the solutions were considerably reduced by employing space-time variational formulations rather than using semidiscretizations. There exist several adaptive algorithms based on space-time FEM. For example, tent-pitching strategies [1, 58, 70, 101] where space-time FEM is employed to build unstructured meshes. In some other algorithms, the authors assume a space-time tensor-product of the trial and test functions. Then, employing discontinuous-in-time basis functions, the space-time FEM can be reinterpreted as a time-marching scheme [127, 129]. Here, the approximation orders (in space and time) as well as the mesh size and the time-step size can be adapted [130, 147].

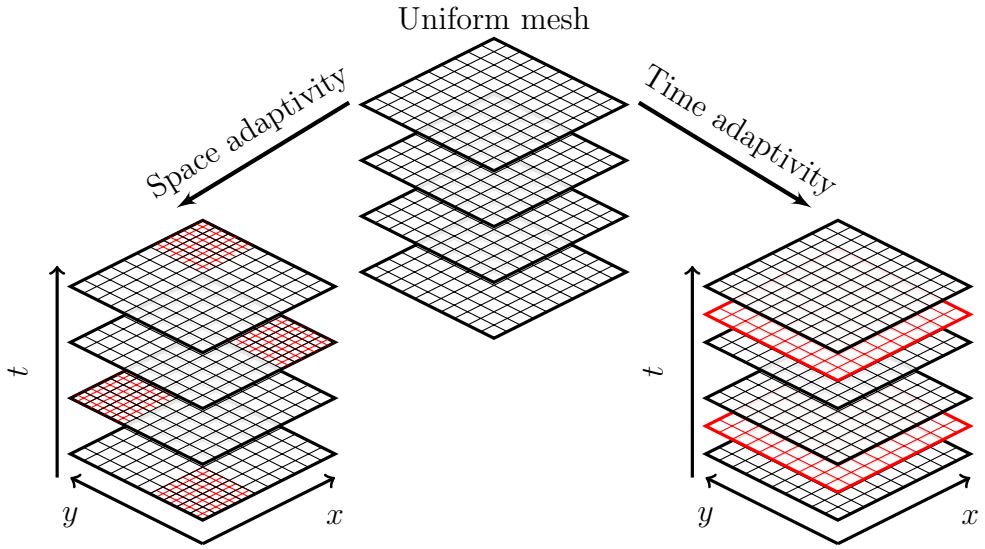



Figure 1.1.: Adaptivity of time-dependent PDEs employing the MoL.

Similarly, to perform GOA for time-dependent PDEs [11, 19, 128], we also require a space-time variational formulation of the problem as in space-time FEM. Here, we represent the error in the QoI as an integral over the whole space-time domain employing the errors of the primal and dual problems. This representation formula is subsequently bounded as a sum of local element contributions, which we use for adaptivity. A full space-time variational formulation [8, 54, 102] allows such representation. Based on this formulation, different goal-oriented adaptive strategies have been developed for PDEs in time-domain. In [9, 10, 12], goal-oriented adaptive strategies are explained for wave propagation phenomena; in [28, 54, 116, 117, 128] for parabolic problems; in [141, 142] the error representation is derived for structural transient dynamics; and finally, in [33, 131, 136, 149] the goal-oriented approach is extended to non-linear problems.

There are several concerns related to GOA in time-domain that we address in this dissertation. One of them is related to the dual problem that is also a time-dependent PDE but running backwards in time. Most authors select discontinuous-in-time test functions to discretize both primal and dual problems. This selection decouples the resulting systems and we can solve them as time-marching schemes [37, 126, 144]. However, as the dual problem runs backwards in time, the goal-oriented adaptive process involves solving two problems running in opposite directions in time. This fact implies that we need to store all the information coming from the primal and dual problems in order to estimate the error contributions and perform the adaptivity.

On the other hand, to discretize both primal and dual problems using the MoL, it is essential to know its equivalence with the corresponding space-time FEM (see Figure 1.2). However, the most used space-time variational formulations for PDEs lead to implicit methods in time when they are thought as time marching schemes. For that reason, most time-domain goal-oriented adaptive processes employ implicit methods in time. It is well known that some low-order space-time FEM are algebraically equivalent to some semidiscretizations [9]. For example, the discontinuous-in-time Galerkin (dG) method using piecewise constant functions (usually denoted by dG(0)) leads to the Backward Euler method. The continuous-in-time Petrov-Galerkin (cPG) method with linear trial and constant test functions (denoted cPG(1)) is equivalent to the Crank-Nicolson method. Recently, higher order dG(r)- and cPG(r)-methods have been developed and analyzed for parabolic and hyperbolic problems [2, 61, 87, 93]. Finally, authors in [33] expressed some multi-step implicit-explicit (IMEX) schemes as Galerkin methods in time for PDEs by using specific quadrature rules for time integration. Our focus is to design goal-oriented adaptive algorithms employing explicit methods in time since in many instances they can be computationally cheaper than implicit ones. To achieve that goal, we need to derive a variational formulation for explicit methods in time.

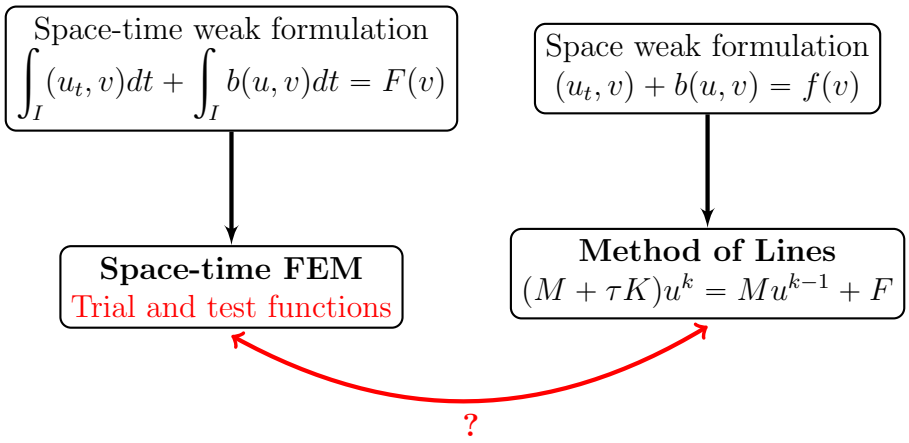



Figure 1.2.: Equivalence between the space-time FEM and the MoL.

There also exist variational formulations of time marching schemes in the context of ODEs [22]. Delfour et. al. [42, 43] and Hulme [86] showed that it is possible to obtain classical schemes like RK methods by employing Galerkin methods for initial value problems together with quadrature formulas. Estep and French [62, 63] derived error bounds for the continuous and discontinuous Galerkin methods to efficiently adapt the time step size. More recently, Estep

and Stuart studied the dynamical behavior of discontinuous Galerkin methods for ODEs in [64] and Tang et. al. provided in [134] a unified framework of finite element methods in time. There also exist some works describing variational formulations of explicit methods for ODEs. In [150], the authors derived some low order explicit RK methods by selecting specific quadrature rules and test functions. However, they claimed that with the formulation they proposed, it is not possible to reproduce second-order RK methods with null weights in the Butcher tableau ($b_j = 0$). In [38], Collins et. al. also proposed a variational formulation for explicit schemes by inserting some operators in the equation together with quadrature rules. They also derived an *a posteriori* error estimation of explicit schemes in the goal-oriented approach. In this case, the errors committed by using the operators and the quadratures must be included in the error representation. In both settings [38, 150], it is possible to recover some explicit RK methods but only on a case-by-case basis and not in a constructive way.

Finally, we also address the fact that if the numerical method employed for the discretization of a PDE is unstable, adaptive algorithms would introduce refinements in unnecessary places. To overcome these problem, it is desirable to employ stable discretization methods. There is a consistent literature on stabilized methods for PDEs [25, 79, 82], specially for advection-diffusion problems where it is well known that the lack of stability is the main issue to overcome. Several discretization techniques are considered within the wide-class of residual minimization methods [103, 104]. These include: the least-squares FEM [21, 83], variational stabilization methods [32, 36] or the Discontinuous Petrov-Galerkin (DPG) method with optimal test functions [47]. In here, we approach the residual minimization using a saddle point (mixed) formulation.

The DPG method with optimal test functions was proposed by Prof. Demkowicz and Gopalakrishnan in 2010 [45, 46, 50, 151] and it has been applied to a wide variety of problems [23, 29, 49, 71, 72]. The key idea of the DPG method is to construct an approximation to the optimal test functions "on the fly", element by element and automatically guarantee the numerical stability of challenging computational problems. In particular, it aims to obtain a practical approach to solve the mixed system derived from the residual minimization method (which is at least two times larger than the original system of equations arising from a standard Galerkin method) by breaking the test spaces at the expense of introducing a hybrid formulation. Recently, the dual of the DPG method has been introduced, the so-called DPG* method [48, 92]. In the latter, another saddle-point problem is solved with a QoI in the right-hand-side. This new methodology allows to perform goal-oriented adaptive refinements employing DPG methods.

Together with residual minimization methods like DPG, we will consider the combination of two other discretization methods: IGA and Alternating Direction Implicit (ADI) methods. IGA is a modern method for performing FEM simulations with B-splines and NURBS [39, 81] which results in a smooth, higher order approximation of the solution. The IGA-FEM has multiple applications for simulations of time-dependent problems [13, 27, 69, 78]. The ADI methods [20, 74, 145] are popular for performing finite difference simulations on regular grids. This method introduces intermediate time steps, and the spatial differential operator splits into the $x$, $y$, and $z$ components. As a result of this operation, on the left-hand side, we only deal with derivatives in one direction, while the rest of the operator is on the right-hand side. The resulting system of linear equations has a multi-diagonal form, so its factorization is possible with a linear $\mathcal{O}(N)$ computational cost [68, 99, 100] being $N$ the number of Degrees of Freedom (DoF).

## 1.2. Main contribution

In this dissertation, we address the aforementioned issues related to time-domain GOA, summarizing the main results from our publications [97, 98, 105, 106, 107, 108].

In [108], we propose a constructive method to derive variational formulations of explicit RK schemes of any stage for linear parabolic PDEs. First, we derive a discontinuous-in-time Petrov-Galerkin (dPG) formulation of the problem where we define the jumps of the solution using a downwind approximation across each time interface. We prove that by selecting piecewise-constant trial and test functions, we recover the Forward Euler method. We allow different spatial discretizations per time interval and in order to obtain square mass matrices, we displace in time the spatial discrete spaces of the test space with respect to the trial space. For a general number of RK stages, we define families of piecewise polynomials and we substitute them into the variational formulation. Treating the coefficients of the polynomials as unknowns, we obtain integrals of the products of trial and test functions. Then, we establish some conditions that these time integrals must satisfy. First, we state the necessary orthogonality conditions needed to obtain an explicit method. We also define non-orthogonality conditions by matching the remaining integrals with the entries of the Butcher's tableau that define the RK methods. Finally, performing analytic integration, we obtain a system of nonlinear equations. By solving this system, we produce the coefficients of the trial and test functions for any stage RK method. We define the corresponding trial and test spaces as the span of these functions. As we fully define the trial and test spaces, we can naturally represent the error in the QoI

in the same way as for implicit methods. Such error representation allows us to design explicit-in-time goal-oriented adaptive algorithms.

In [106], we derive a space-time error representation of the QoI and develop a goal-oriented adaptive process in space employing explicit methods in time following the ideas presented in [108]. We allow dynamic meshes in space, that is, we consider different spatial discretizations per time interval. Then, in the dPG formulation, we select piecewise constant trial and test functions, which lead to the Forward Euler method in time. To discretize the dual problem, we shift the trial and test spaces from the primal problem and we also obtain the Forward Euler method but running backwards in time. As we are solving the primal and dual problems with explicit methods in time, the Courant-Friedrichs-Lewy (CFL) condition [132] must be satisfied to ensure the stability of the discretization. Our adaptive strategy ensures the CFL condition by construction, as we start with a time step size that is in the limit of the CFL condition. We further assume (as it occurs in many applications) that once the CFL condition has been satisfied, most of the error is due to an inadequate space discretization. This assumption is verified in the examples considered in this dissertation. Our adaptive scheme focuses on performing optimal goal-oriented space refinements followed by a simple refinement condition in time intended to recover the CFL condition after those space refinements. Therefore, employing the error in space of the primal and dual problems, we obtain the error representation that drives the GOA in space.

In [107], following the ideas presented in [40, 41], we define a pseudo-dual problem that, as the primal problem, runs forward in time. Then, we define a forward-in-time goal-oriented adaptive process. In the proposed algorithm, we solve the primal and dual problems at a fixed time step to adapt the corresponding spatial mesh before we move to the next time step. This is possible because both primal and dual problems run in the same direction in time. This approach overcomes the memory storage issue of classical algorithms. We can perform the adaptivity as we compute the solution, thus we do not need to store the solution at every time step for each iteration. However, this algorithm only works in some specific problem configurations: for example, in diffusion problems where the QoI is localized in some area of the spatial domain and has support in the whole time interval. For advection-diffusion problems and for problems where the QoI is placed in a reduced late time interval, the pseudo-dual problem we define is not generally well defined. In general, it is not possible to define a dual problem running forward in time that captures the necessary information from the future time steps. Nevertheless, the method works properly in several relevant instances and could be useful in some engineering problems when some features of the solution are known beforehand. As an alternative for advection-

diffusion problems, we also propose a hybrid algorithm that solves the classical dual problem backwards in time once. Then, we set the initial condition of the pseudo-dual problem as the initial condition of the classical dual problem. Finally, the adaptive process is performed forwards in time. In both algorithms proposed herein, we only need to store the solutions of the primal and dual problems at one time step to perform the adaptivity.

In [105], we extend the new error representations developed in [40, 41] for multidimensional Helmholtz problems to the 1D time-domain wave equation. We select an alternative bilinear form by modifying from the classical one the terms coming from the second order time derivative. By doing so, we build a pseudo-dual problem by modifying the classical dual problem in a way that we obtain another wave propagation problem with better stability properties. We compare the classical upper bounds of the error in the QoI with the new ones, observing that the new bounds are sharper for the 1D wave equation. The resulting method provides a better guidance criterion for the adaptive process.

Finally, in [97, 98], we propose a method that brings together the benefits from ADI methods, residual minimization and IGA. We call our method Isogeometric Residual Minimization (iGRM) with direction splitting and we seek to obtain, in an efficient way, stable and accurate solutions of time-dependent problems. In order the stabilize the time-dependent advection-diffusion simulations, we perform the following steps: First, we apply several second order ADI time integration schemes. Particularly, we employ the Peaceman-Reachford [118] and Strang [132] splitting schemes in 2D+time and the Douglas-Gunn [55, 56] in 3D+time. Second, we derive a variational formulation in space and we stabilize the time-marching scheme by applying a residual minimization method to each time step. Finally, we perform the spatial discretization with IGA using tensor product B-spline basis functions. Similarly as in the DPG method, at each time-time step we solve a saddle point problem. However, we do not consider broken test spaces and hybridization. Here, we preserve the Kronecker product structure of the entire system, even after applying the residual minimization method at each time step. We can factorize this kind of matrix with a linear computational cost $\mathcal{O}(N)$ solver. We show via numerical results in 2D and 3D+time that the proposed method overcomes the stability issues arising from classical Galerkin methods for the time-domain advection-diffusion equation. The proposed iGRM method is the first step towards the derivation of a goal-oriented adaptive strategy (similarly as in the DPG* method) for time-dependent PDEs employing stabilized time-marching schemes that we will consider in future works.

## 1.3. Outline

The remainder of the dissertation is organized as follows: Chapter 2 describes the model problem we employ in this dissertation and summarizes the classical dG and cPG formulations that lead to implicit methods in time. Chapter 3 demonstrates that explicit RK methods can be reinterpreted as dPG methods. For any explicit RK method, we build the corresponding trial and test basis functions employing exact integration in time. In Chapter 4, we propose an explicit-in-time goal-oriented adaptive algorithm employing the Forward Euler method that adapts locally the time grid based on the CFL condition. Chapter 5 explains the proposed forward-in-time goal-oriented adaptive process for a fixed time grid and the hybrid algorithm for advection-diffusion problems. Chapter 6 derives the unconventional error representation applied to the 1D wave equation and a space-time tensor product goal-oriented adaptive algorithm. Chapter 7 describes iGRM method with direction splitting we propose together with 2D and 3D+time numerical results. Chapter 8 is devoted to the conclusions and future works and Chapter 9 provides a list of main achievements. This work also contains three appendices. Appendix A expresses in matrix form the nonlinear system of equations we need to solve to obtain any explicit RK method and provides a MATLAB code to solve it. Appendix B discusses the stability constraint for the explicit-in-time goal-oriented adaptive algorithm and defines the discrete bilinear form for the adjoint problem. Finally, Appendix C recalls the basic ideas of residual minimization methods and $LU$ factorization of matrices with Kronecker product structure.

# 2. Variational formulations for implicit-in-time methods

In this chapter, we introduce basic ideas about space-time variational formulations that lead to implicit methods in time. First, we state both the strong and weak formulations of the model problem we employ to develop the theory presented in this dissertation. Then, we state the classical discontinuous-in-time Galerkin (dG) formulation and its equivalence with the (implicit-in-time) Backward Euler method for the lowest order case. Finally, we show that we need a continuous-in-time Petrov-Galerkin (cPG) formulation to obtain an implicit method like Crank-Nicolson.

## 2.1. Model problem and variational formulation

Let $\Omega$ and open bounded subset of $\mathbb{R}^d$ with $d \in \{1, 2, 3\}$ and $I = (0, T] \subset \mathbb{R}$, we consider the *linear advection-diffusion equation*

$$\begin{cases} u_t - \nabla \cdot (\nu \nabla u) + \beta \cdot \nabla u = f & \text{in } \Omega \times I, \\ u = 0 & \text{on } \partial\Omega \times I, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \tag{2.1}$$

where $u_t := \partial u / \partial t$ and $\partial\Omega$ denotes the boundary of the spatial domain $\Omega$. For arbitrary Dirichlet boundary conditions, we can modify the source term accordingly, thus making (2.1) a general statement.

The solution $u(\mathbf{x}, t)$ represents the temperature distribution in a body. The source term $f(\mathbf{x}, t)$, the initial condition $u_0(\mathbf{x})$, the diffusivity tensor $\nu(\mathbf{x})$ and the velocity field $\beta(\mathbf{x})$ are given data. We assume that $\beta(\mathbf{x})$ is a bounded divergence-free vector field

$$\nabla \cdot \beta = 0,$$

and $\nu(\mathbf{x})$ is a bounded above and strictly positive symmetric second order tensor. For simplicity, we omit the spatial dependence of the functions, that is, we write $u(t)$ instead of $u(\mathbf{x}, t)$.

We need a full space-time variational formulation of problem (2.1) [9, 54] in order to properly define a dual problem and to subsequently represent the error in the Quantity of Interest (QoI) as an integral over the whole space-time domain.

Weak enforcement of the diffusion term and strong enforcement of the boundary conditions in (2.1) implies the following Hilbert space

$$V := H_0^1(\Omega) = \{u \in L^2(\Omega) \mid \nabla u \in L^2(\Omega),\ u = 0 \text{ on } \partial\Omega\}.$$

Therefore, a simple way to ensure sufficient integrability of the weak formulation is to impose that $u_t$ and $f$ should belong to the dual space of $V$, i.e., $V' = H^{-1}(\Omega)$. Now, we introduce the following test space

$$\mathcal{V} := L^2(I;V) = \left\{ u : I \longrightarrow V \,|\, u \,\text{is}\, V\text{-measurable and } \int_I ||u(t)||_V^2 dt < +\infty \right\},$$

which is the Bochner space of all integrable functions in time that take values in $V$ [125]. We denote the dual space of $\mathcal{V}$ as $\mathcal{V}' := L^2(I;V')$.

For the solution, we need $u \in \mathcal{V}$ and $u_t \in \mathcal{V}'$. So we define the following trial space

$$\mathcal{U} := \{u \in \mathcal{V} \mid u_t \in \mathcal{V}'\},$$

which is a subspace of functions that are $H^1$ in time. Therefore, all functions in $\mathcal{U}$ are globally continuous in time.

Now, we multiply the advection-diffusion equation by the test functions $v \in \mathcal{V}$ and we integrate over the entire domain $\Omega \times I$. We also impose the initial condition in weak form. Finally, assuming that $f \in \mathcal{V}$ and $u_0 \in L^2(\Omega)$, the weak solution of problem (2.1) belongs to $\mathcal{U}$ and satisfies

$$\begin{aligned} \int_I \langle u_t, v\rangle\, dt + \int_I (\nu\nabla u, \nabla v) dt + \int_I (\beta\cdot\nabla u, v) dt &= \int_I \langle f, v\rangle\, dt,\ \forall v \in \mathcal{V}, \\ (u(0), \hat{v}) &= (u_0, \hat{v}),\ \forall \hat{v} \in L^2(\Omega). \end{aligned} \tag{2.2}$$

Here, $\langle\cdot,\cdot\rangle$ denotes the duality pairing between the spaces $V$ and $V'$, $(\cdot,\cdot)$ is the inner product in $L^2(\Omega)$, and we define

$$B(u,v) := \int_I \langle u_t, v\rangle\, dt + \int_I (\nu\nabla u, \nabla v) dt + \int_I (\beta\cdot\nabla u, v) dt. \tag{2.3}$$

## 2.2. Classical discretizations

First, we define a partition of the time interval $\bar{I} = [0,T]$ as

$$0 = t_0 < t_1 < \ldots < t_{m-1} < t_m = T, \tag{2.4}$$

and we denote by $I_k = (t_{k-1}, t_k)$, $\tau_k := t_k - t_{k-1},\ \forall k = 1, \dots, m$ and $\tau := \max_{1 \leqslant k \leqslant m} \tau_k$.

In space, we can employ either a Finite Element Method (FEM) or a Spectral Element Method (SEM) [119]. We allow dynamic meshes in space so we define a finite dimensional subspace of $V$ for each time step, i.e., $V_h^k \subset V,\ \forall k = 0, \dots, m$, where $h$ is the largest element diameter of each dynamic mesh.

## 2.2.1. Discontinuous-in-time Galerkin (dG) formulation

For the classical dG formulation, we select the following semidiscrete space

$$\mathcal{V}_\tau^r := \{v \in L^2(I; V) \,|\, v_{|_{(t_{k-1}, t_k]}} \in P_r((t_{k-1}, t_k]; V),\ \forall k = 1, \dots, m,\ v(0) \in V\}, \tag{2.5}$$

where $\mathcal{V}_\tau^r \subset \mathcal{V}$ and $P_r(I_k; V)$ is the space of all polynomials with degree less or equal than $r$ on the interval $I_k$ taking values in $V$.

The functions in $\mathcal{V}_\tau^r$ could be discontinuous at the end of each time step $t_k$, so we define the jump of the function $v$ at $t_k$ as

$$[\![v]\!]^k := v(t_k^+) - v(t_k^-), \tag{2.6}$$

where $v(t_k^\pm) := \lim_{\epsilon \longrightarrow 0^+} v(t_k \pm \epsilon)$.

We define a semidiscrete dG formulation of (2.2) as

$$\begin{cases} \text{Find } u_\tau \in \mathcal{V}_\tau^r \text{ such that} \\ B_{DG}^+(u_\tau, v_\tau) = F(v_\tau),\ \forall v_\tau \in \mathcal{V}_\tau^r, \end{cases} \tag{2.7}$$

where

$$\begin{aligned} B_{DG}^+(u, v) :&= \sum_{k=1}^m \int_{I_k} \Big( \langle u_t, v \rangle + (\nu \nabla u, \nabla v) + (\beta \cdot \nabla u, v) \Big) dt \\ &+ \sum_{k=1}^m \left( [\![u]\!]^{k-1}, v(t_{k-1}^+) \right) + (u(0^-), v(0^-)), \\ F(v) :&= \sum_{k=1}^m \int_{I_k} \langle f, v \rangle \, dt + (u_0, v(0^-)). \end{aligned} \tag{2.8}$$

The jumps in (2.8) come from integrating by parts twice in time and employing an upwind approximation of the solution at each time interface. Notice that we can express

$$B_{DG}^+(u, v) = B(u, v) + \sum_{k=1}^m \left( [\![u]\!]^{k-1}, v(t_{k-1}^+) \right) + (u(0^-), v(0^-)). \tag{2.9}$$

Therefore, the bilinear form $B^+_{DG}(\cdot,\cdot)$ is a generalization of $B(\cdot,\cdot)$ for discontinuous-in-time functions. As the solution of (2.2) is continuous in time, then $[\![u]\!]^{k-1} = 0,\ \forall k = 1,\dots,m$, so it also satisfies problem (2.7).

Now, for the fully discrete version, we define the following discrete space

$$\mathcal{V}^r_{\tau h} := \{v \in L^2(I;V) \,|\, v_{|_{(t_{k-1},t_k]}} \in P_r((t_{k-1},t_k];V^k_h),\ \forall k = 1,\dots,m,\ v(0) \in V^0_h\}, \tag{2.10}$$

and we state that a fully discrete dG formulation of (2.2) [123, 128] is

$$\begin{cases} \text{Find } u_{\tau h} \in \mathcal{V}^r_{\tau h} \text{ such that} \\ B^+_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}^r_{\tau h}. \end{cases} \tag{2.11}$$

#### 2.2.1.1. Equivalence with Backward Euler method

As the test functions in (2.11) are discontinuous-in-time and have local support in $I_k$, we can split the formulation into $m$ local-in-time problems and solve it using a time-marching scheme.

First, we select trial and test functions that can be expressed as a Cartesian product of functions in time and space. In (2.10), if we select constant functions in time ($r = 0$), we can express the solution of (2.11) as

$$u_{\tau h}(t) = \sum_{k=0}^{m} u^k_h \varphi^k(t), \tag{2.12}$$

where $u^k_h \in V^k_h,\ \forall k = 0,\dots,m$, and the basis functions in time are

$$\varphi^0(t) = \begin{cases} 1, & t = t_0, \\ 0, & \text{elsewhere}, \end{cases} \qquad \varphi^k(t) = \begin{cases} 1, & t \in (t_{k-1}, t_k], \\ 0, & \text{elsewhere}. \end{cases} \tag{2.13}$$

We select the following test functions

$$v^k_h \varphi^k(t),\ \forall k = 0,\dots,m, \tag{2.14}$$

where $v^k_h \in V^k_h,\ \forall k = 0,\dots,m$. Figure 2.1 shows the trial and test basis functions for this particular case.

We commit a slight abuse of notation by omitting the constants in (2.12) because we can express each function $u^k_h$ as a linear combination of basis functions in $V^k_h,\ \forall k = 0,\dots,m$.

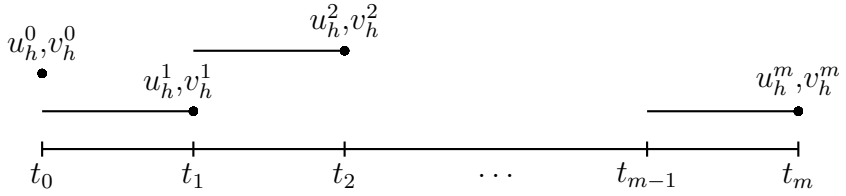



Figure 2.1.: Trial and test functions for the classical dG formulation ($r = 0$).

*Theorem* 2.1. (Backward Euler method) Selecting trial and test functions as in (2.12) and (2.14), problem (2.11) leads to the following scheme

$$\begin{cases} \text{Find } u_h^0 \in V_h^0 \text{ and } u_h^k \in V_h^k,\ \forall k = 1, \dots, m, \text{ such that} \\ \\ \left(u_h^k, v_h^k\right) + \tau_k \left(\beta \cdot \nabla u_h^k, v_h^k\right) + \tau_k \left(\nu \nabla u_h^k, \nabla v_h^k\right) \\ \qquad = \left(u_h^{k-1}, v_h^k\right) + \displaystyle\int_{I_k} \left\langle f, v_h^k \right\rangle dt,\ \forall v_h^k \in V_h^k, \\ \\ \left(u_h^0, v_h^0\right) = \left(u_0, v_h^0\right),\ \forall v_h^0 \in V_h^0, \end{cases} \tag{2.15}$$

which is an implicit method that is a variant of the Backward Euler method in time.

*Proof.* Substituting (2.12) and the first test function of (2.14), $v_{\tau h}(t) = v_h^0 \varphi^0(t)$, into (2.11), we obtain

$$(u_h^0, v_h^0) = (u_0, v_h^0),\ \forall v_h^0 \in V_h^0,$$

which is the $L^2$-projection in space of the initial condition $u_0$ on $V_h^0 \subset V$.

Then, for $k = 1, \dots, m$, each test function of the form $v_{\tau h}(t) = v_h^k \varphi^k(t)$ has local support in $(t_{k-1}, t_k]$, so problem (2.11) reads

$$\begin{aligned} \int_{I_k} \langle u_{\tau h,t}, v_{\tau h} \rangle \, dt + \int_{I_k} (\nu \nabla u_{\tau h}, \nabla v_{\tau h}) dt + \int_{I_k} (\beta \cdot \nabla u_{\tau h}, v_{\tau h}) dt \\ + (\llbracket u_{\tau h} \rrbracket^{k-1}, v_{\tau h}(t_{k-1}^+)) = \int_{I_k} \langle f, v_{\tau h} \rangle \, dt. \end{aligned}$$

Now, as both trial and test functions are piecewise constant in time, each term of the previous formula becomes

$$\int_{I_k} \langle u_{\tau h,t}, v_{\tau h} \rangle \, dt = (u_h^k, v_h^k) \int_{I_k} \left( \frac{\partial}{\partial t} \varphi^k \right) \varphi^k dt = 0,$$

$$\int_{I_k} (\nu\nabla u_{\tau h}, \nabla v_{\tau h})dt = (\nu\nabla u_h^k, \nabla v_h^k)\int_{I_k} \varphi^k\varphi^k dt = \tau_k(\nu\nabla u_h^k, \nabla v_h^k),$$

$$\int_{I_k} (\beta\cdot\nabla u_{\tau h}, v_{\tau h})dt = (\beta\cdot\nabla u_h^k, v_h^k)\int_{I_k} \varphi^k\varphi^k dt = \tau_k(\beta\cdot\nabla u_h^k, v_h^k),$$

$$([\![u_{\tau h}]\!]^{k-1}, v_{\tau h}(t_{k-1}^+)) = (u_h^k\varphi^k(t_{k-1}^+) - u_h^{k-1}\varphi^{k-1}(t_{k-1}^-), v_h^k\varphi^k(t_{k-1}^+)) = (u_h^k - u_h^{k-1}, v_h^k),$$

$$\int_{I_k} \langle f, v_{\tau h}\rangle\, dt = \int_{I_k} \left\langle f, v_h^k\right\rangle dt.$$

Finally, we obtain

$$\left(u_h^k, v_h^k\right) + \tau_k\left(\nu\nabla u_h^k, \nabla v_h^k\right) + \tau_k(\beta\cdot\nabla u_h^k, v_h^k) = \left(u_h^{k-1}, v_h^k\right) + \int_{I_k}\left\langle f, v_h^k\right\rangle dt.$$

□

### 2.2.2. Continuous-in-time Petrov-Galerkin (cPG) formulation

In the classical cPG formulation, we consider the same semidiscrete test space $\mathcal{V}_\tau^r$ we defined for the dG formulation in (2.5). Then, we define the following semidiscrete trial space

$$\mathfrak{U}_\tau^r := \{w \in C(\bar{I}; V)\,|\, w_{|_{\bar{I}_k}} \in P_r(\bar{I}_k; V),\ \forall k = 1, \dots, m\}, \tag{2.16}$$

where $\mathfrak{U}_\tau^r \subset \mathcal{U}$ and the functions in $\mathfrak{U}_\tau^r$ are globally continuous in time.

Now, we define a semidiscrete cPG formulation of (2.2) as

$$\begin{cases} \text{Find } u_\tau \in \mathfrak{U}_\tau^r \text{ such that} \\ B_{CG}(u_\tau, v_\tau) = F(v_\tau),\ \forall v_\tau \in \mathcal{V}_\tau^{r-1}, \end{cases} \tag{2.17}$$

where

$$B_{CG}(u, v) = B(u, v) + (u(0), v(0^-)).$$

Notice that in (2.17), to obtain a square system, the polynomials in the test functions are one degree smaller than in the trial functions.

In order to define the fully discrete version of (2.17), we follow a similar construction as in [128]. First, we consider a basis of the polynomials of order $r$ per time interval

$$\psi_0^k(t), \dots, \psi_r^k(t),\ \forall k = 1, \dots, m,$$

satisfying the following condition

$$\psi_0^k(t_{k-1}) = 1,\ \psi_0^k(t_k) = 0,\ \psi_i^k(t_{k-1}) = 0,\ \forall i = 1, \dots, r,\ \psi_r^k(t_k) = 1. \tag{2.18}$$

Then, we set

$$\mathfrak{U}_{\tau h}^{r,k} = \operatorname{span}\left\{ w_{h,0}^k \psi_0^k,\ w_{h,i}^k \psi_i^k \mid w_{h,0}^k \in V_h^{k-1},\ w_{h,i}^k \in V_k^k,\ \forall i = 1, \dots, r \right\},$$

and to ensure the global continuity of the trial functions apart from (2.18), we also need to enforce

$$w_{h,r}^{k-1} = w_{h,0}^k,\ \forall k = 2, \dots, m. \tag{2.19}$$

Finally, we define the fully discrete trial space as

$$\mathfrak{U}_{\tau h}^r := \{ w \in C(\bar{I}; V) \mid w_{|\bar{I}_k} \in \mathfrak{U}_{\tau h}^{r,k},\ \forall k = 1, \dots, m \}, \tag{2.20}$$

and the fully discrete version of (2.17) as

$$\begin{cases} \text{Find } u_{\tau h} \in \mathfrak{U}_{\tau h}^r \text{ such that} \\ B_{CG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}_{\tau h}^{r-1}, \end{cases} \tag{2.21}$$

where $\mathcal{V}_{\tau h}^r$ is the discrete space defined in (2.10).

#### 2.2.2.1. Equivalence with Crank-Nicolson method

In (2.20), if we select piecewise linear functions in time ($r = 1$), we can express the solution of (2.21) as

$$u_{\tau h}(t) = \sum_{k=1}^{m} u_{h,0}^k \psi_0^k(t) + u_{h,1}^k \psi_1^k(t), \tag{2.22}$$

where $u_{h,0}^k \in V_h^{k-1},\ u_{h,1}^k \in V_h^k,\ \forall k = 1, \dots, m$, and we define the basis functions in time as

$$\psi_0^k(t) = \begin{cases} \dfrac{t_k - t}{\tau_k}, & t \in [t_{k-1}, t_k], \\ 0, & \text{elsewhere}, \end{cases} \qquad \psi_1^k(t) = \begin{cases} \dfrac{t - t_{k-1}}{\tau_k}, & t \in [t_{k-1}, t_k], \\ 0, & \text{elsewhere}. \end{cases} \tag{2.23}$$

Now we denote

$$u_h^k := u_{\tau h}(t_k),\ \forall k = 0, \dots, m,$$

so from conditions (2.18) and (2.19), we have the following equalities

$$u_h^0 = u_{h,0}^1,\ \ u_h^{k-1} = u_{h,1}^{k-1} = u_{h,0}^k,\ \forall k = 2, \dots, m,\ \ u_h^m = u_{h,1}^m. \tag{2.24}$$

Finally, we select test functions defined in (2.14). Figure 2.2 shows the trial and test basis functions for this particular case.

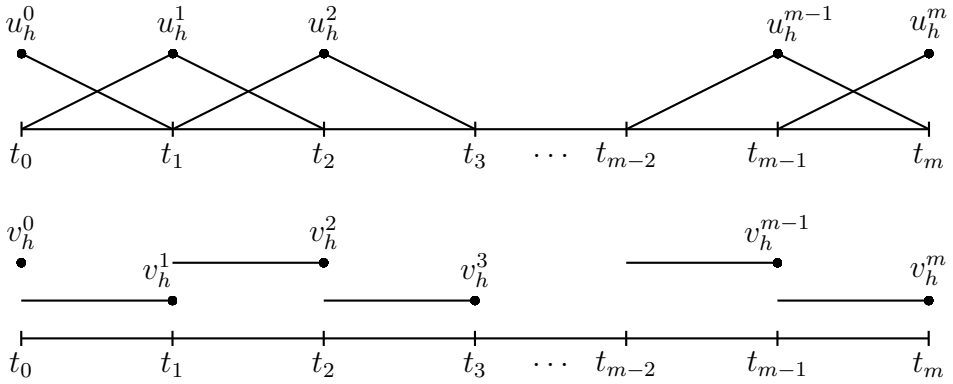



Figure 2.2.: Trial and test functions for the classical cPG formulation ($r = 1$).

*Theorem* 2.2. (Crank-Nicolson method) Selecting trial and test functions as in (2.22) and (2.14), problem (2.21) leads to the following scheme

$$\begin{cases} \text{Find } u_h^0 \in V_h^0 \text{ and } u_h^k \in V_h^k,\ \forall k = 1, \dots, m, \text{ such that} \\ \\ (u_h^k, v_h^k) + \dfrac{\tau_k}{2}(\nu \nabla u_h^k, \nabla v_h^k) + \dfrac{\tau_k}{2}(\beta \cdot \nabla u_h^k, v_h^k) = (u_h^{k-1}, v_h^k) \\ \qquad - \dfrac{\tau_k}{2}(\nu \nabla u_h^{k-1}, \nabla v_h^k) - \dfrac{\tau_k}{2}(\beta \cdot \nabla u_h^{k-1}, v_h^k) + \displaystyle\int_{I_k} \left\langle f, v_h^k \right\rangle dt,\ \forall v_h^k \in V_h^k, \\ \\ \left(u_h^0, v_h^0\right) = \left(u_0, v_h^0\right),\ \forall v_h^0 \in V_h^0, \end{cases} \tag{2.25}$$

which is an implicit method that is a variant of the Crank-Nicolson method in time.

*Proof.* First, we substitute (2.22) and the first test function of (2.14), $v_{\tau h}(t) = v_h^0 \varphi^0(t)$, into (2.21). From (2.24), as $u_h^0 = u_{h,0}^1$, we obtain

$$(u_h^0, v_h^0) = (u_0, v_h^0),\ \forall v_h^0 \in V_h^0.$$

Now, as each test function $v_{\tau h}(t) = v_h^k \varphi^k(t)$ has local support in $(t_{k-1}, t_k]$, $\forall k = 1, \dots, m$, problem (2.21) reads

$$\int_{I_k} \langle u_{\tau h,t}, v_{\tau h} \rangle \, dt + \int_{I_k} (\nu \nabla u_{\tau h}, \nabla v_{\tau h}) dt + \int_{I_k} (\beta \cdot \nabla u_{\tau h}, v_{\tau h}) dt = \int_{I_k} \langle f, v_{\tau h} \rangle \, dt.$$

Then, each term of the previous equation becomes

$$\begin{aligned}
\int_{I_k} \langle u_{\tau h,t}, v_{\tau h}\rangle \, dt &= (u^k_{h,0}, v^k_h)\int_{I_k} (\frac{\partial}{\partial t}\psi^k_0)\varphi^k dt + (u^k_{h,1}, v^k_h)\int_{I_k} (\frac{\partial}{\partial t}\psi^k_1)\varphi^k dt \\
&= -(u^k_{h,0}, v^k_h) + (u^k_{h,1}, v^k_h), \\
\int_{I_k} (\nu\nabla u_{\tau h}, \nabla v_{\tau h}) dt &= (\nu\nabla u^k_{h,0}, \nabla v^k_h)\int_{I_k} \psi^k_0\varphi^k dt + (\nu\nabla u^k_{h,1}, \nabla v^k_h)\int_{I_k} \psi^k_1\varphi^k dt \\
&= \frac{\tau_k}{2}(\nu\nabla u^k_{h,0}, \nabla v^k_h) + \frac{\tau_k}{2}(\nu\nabla u^k_{h,1}, \nabla v^k_h), \\
\int_{I_k} (\beta\cdot\nabla u_{\tau h}, v_{\tau h}) dt &= (\beta\cdot\nabla u^k_{h,0}, v^k_h)\int_{I_k} \psi^k_0\varphi^k dt + (\beta\cdot\nabla u^k_{h,1}, v^k_h)\int_{I_k} \psi^k_1\varphi^k dt \\
&= \frac{\tau_k}{2}(\beta\cdot\nabla u^k_{h,0}, v^k_h) + \frac{\tau_k}{2}(\beta\cdot\nabla u^k_{h,1}, v^k_h), \\
\int_{I_k} \langle f, v_{\tau h}\rangle \, dt &= \int_{I_k} \left\langle f, v^k_h\right\rangle dt.
\end{aligned}$$

Finally, from (2.24),we obtain

$$\begin{aligned}
&(u^k_h, v^k_h) + \frac{\tau_k}{2}(\nu\nabla u^k_h, \nabla v^k_h) + \frac{\tau_k}{2}(\beta\cdot\nabla u^k_h, v^k_h) \\
&\quad = (u^{k-1}_h, v^k_h) - \frac{\tau_k}{2}(\nu\nabla u^{k-1}_h, \nabla v^k_h) - \frac{\tau_k}{2}(\beta\cdot\nabla u^{k-1}_h, v^k_h) + \int_{I_k} \left\langle f, v^k_h\right\rangle dt.
\end{aligned}$$

□

In conclusion, we have introduced the classical dG and cPG formulations and their equivalence with Backward Euler and Crank-Nicolson methods, respectively. We will employ ideas from both constructions to build variational formulations for explicit-in-time methods in the next chapter.

# 3. Variational formulation for explicit Runge-Kutta (RK) methods

In this chapter, we derive a discontinuous-in-time Petrov-Galerkin (dPG) formulation that is equivalent to explicit RK methods in time. We employ similar ideas to the ones presented in Section 2.2. To facilitate the understanding of the proposed construction, we first derive the Forward Euler method. Then, we construct the subspaces for some two-stage and second order explicit RK methods and finally, we generalize the process to any s-stage explicit RK method. The construction presented in this chapter could be used to represent the error in the Quantity of Interest (QoI) in order to design explicit-in-time goal-oriented adaptive algorithms.

## 3.1. Discontinuous-in-time Petrov-Galerkin (dPG) formulation

First, considering the partition defined in (2.4), we select the semidiscrete test space $\mathcal{V}_\tau^r$ defined in (2.5) and the following semidiscrete trial space

$$\mathcal{U}_\tau^q := \{u \in L^2(I;V) \,|\, u_{|_{[t_{k-1},t_k)}} \in P_q([t_{k-1},t_k)\,;V),\ \forall k = 1,\dots,m,\ u(T) \in V\}. \tag{3.1}$$

Here, the trial functions are right-discontinuous while the test functions are left-discontinuous in time [134] (see Figure 3.1). Therefore, both $\mathcal{U}_\tau^q$ and $\mathcal{V}_\tau^r$ are subspaces of $\mathcal{V}$. However, $\mathcal{U}_\tau^q \not\subset \mathcal{U}$ (because functions in $\mathcal{U}_\tau^q$ are discontinuous) and $\mathcal{U} \not\subset \mathcal{U}_\tau^q$ (since functions in $\mathcal{U}_\tau^q$ are piecewise polynomials).

As we consider discontinuous functions in time and different trial and test spaces, we need a dPG formulation of problem (2.2). First, we integrate by parts in time the bilinear form (2.3) over each subinterval $I_k$

$$\int_{I_k} \Big( -\langle u, v_t\rangle + (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v)\Big)dt + (u(t_k^-), v(t_k^-)) - (u(t_{k-1}^+), v(t_{k-1}^+)).$$

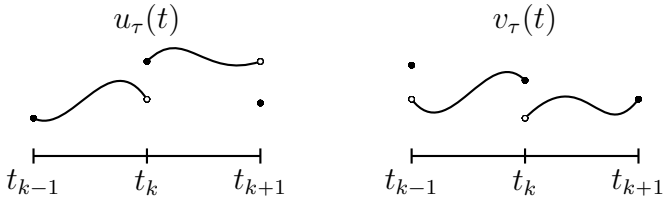



Figure 3.1.: Discontinuous trial and test functions.

At each time interface, instead of performing an upwind approximation of the solution like in the standard discontinuous-in-time Galerkin (dG) approach, we perform a *downwind approximation* in order to obtain an explicit method. That is, we substitute $u(t_k^-)$ by $u(t_k^+)$ in the above formula. Then, integrating by parts again, we obtain

$$\int_{I_k} \Big( \langle u_t, v\rangle + (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v)\Big) dt + (u(t_k^+), v(t_k^-)) - (u(t_k^-), v(t_k^-)),$$

and equivalently

$$\int_{I_k} \Big( \langle u_t, v\rangle + (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v)\Big) dt + (\llbracket u\rrbracket^k, v(t_k^-)), \tag{3.2}$$

where $\llbracket u\rrbracket^k$ are the jumps defined in (2.6).

We define two subspaces $\tilde{\mathcal{U}}_\tau \subset \mathcal{U}_\tau^q$ and $\tilde{\mathcal{V}}_\tau \subset \mathcal{V}_\tau^r$ as

$$\begin{aligned}
\tilde{\mathcal{U}}_\tau &:= \{u \in L^2(I;V) \,|\, u_{|_{[t_{k-1},t_k)}} \in P_q^{inc_u}([t_{k-1},t_k);V),\ \forall k=1,\dots,m,\ u(T)\in V\},\\
\tilde{\mathcal{V}}_\tau &:= \{v \in L^2(I;V) \,|\, v_{|_{(t_{k-1},t_k]}} \in P_r^{inc_v}((t_{k-1},t_k];V),\ \forall k=1,\dots,m,\ v(0)\in V\},
\end{aligned} \tag{3.3}$$

where $P_q^{inc_u}(I_k;V)$ (and $P_r^{inc_v}(I_k;V)$) are (possibly) incomplete spaces of polynomials of degree less than or equal to $q$ (and $r$, respectively) taking values in $V$. We will define explicitly such incomplete spaces of polynomials in the next section.

Now, summing the expression (3.2) over all intervals, and adding the initial condition, we obtain the following dPG formulation of (2.2)

$$\begin{cases} \text{Find } u_\tau \in \tilde{\mathcal{U}}_\tau \subset \mathcal{U}_\tau^q \text{ such that} \\ B_{DG}^-(u_\tau, v_\tau) = F(v_\tau),\ \forall v_\tau \in \tilde{\mathcal{V}}_\tau \subset \mathcal{V}_\tau^r, \end{cases} \tag{3.4}$$

where the bilinear form $B_{DG}^-(\cdot,\cdot)$ is defined by

$$B_{DG}^-(u,v) := B(u,v) + \sum_{k=1}^m \big(\llbracket u\rrbracket^k, v(t_k^-)\big) + (u(0^+), v(0^-)), \tag{3.5}$$

and $F(\cdot)$ is the linear form defined in (2.8). Note that in the classical dG formulation, the jumps of the solution are located at the left endpoint of each temporal interval $I_k$, while in the proposed dPG, the jumps are placed at the right endpoint (see Figure 3.1).

Finally, we introduce two new subspaces $\tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_\tau$ and $\tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_\tau$ and we define the fully discrete problem as

$$\begin{cases} \text{Find } u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_\tau \text{ such that} \\ B^-_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_\tau. \end{cases} \tag{3.6}$$

We will determine subspaces $\tilde{\mathcal{U}}_{\tau h}$ and $\tilde{\mathcal{V}}_{\tau h}$ in the next section so we recover explicit RK methods for (3.6).

## 3.2. Equivalence with explicit RK methods

For simplicity of the construction, in the remainder of this chapter we consider the linear diffusion equation in (2.1). Then, the corresponding bilinear form is

$$B(u,v) := \int_I \langle u_t, v\rangle\, dt + \int_I (\nabla u, \nabla v) dt. \tag{3.7}$$

The extension to advection-diffusion equation (2.1) is straightforward.

### 3.2.1. Forward Euler method

To derive the Forward Euler method, we select in (3.3) piecewise constant functions in time, i.e. $q = r = 0$ and we define the subspaces in (3.6) as

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h} &:= \{u \in L^2(I;V) \,|\, u_{|_{[t_{k-1},t_k)}} \in P_0([t_{k-1},t_k);V_h^{k-1}),\ \forall k = 1,\dots,m,\ u(T) \in V_h^m\}. \\ \tilde{\mathcal{V}}_{\tau h} &:= \{v \in L^2(I;V) \,|\, v_{|_{(t_{k-1},t_k]}} \in P_0((t_{k-1},t_k];V_h^{k}),\ \forall k = 1,\dots,m,\ v(0) \in V_h^0\}. \end{aligned} \tag{3.8}$$

In this particular case, the subspaces defined in (3.3) coincide with the spaces defined in (3.1) and (2.5) with $q = r = 0$, i.e., $\tilde{\mathcal{U}}_\tau = \mathcal{U}_\tau^0$ and $\tilde{\mathcal{V}}_\tau = \mathcal{V}_\tau^0$.

We express the solution of (3.6) as follows (see Figure 3.2)

$$u_{\tau h}(t) = \sum_{k=1}^{m+1} u_h^{k-1} \phi^{k-1}(t), \tag{3.9}$$

where $u_h^{k-1} \in V_h^{k-1},\ \forall k = 1,\dots,m+1$ and the trial functions are

$$\phi^{k-1}(t) = \begin{cases} 1, & t \in [t_{k-1}, t_k), \\ 0, & \text{elsewhere}, \end{cases} \qquad \phi^{m}(t) = \begin{cases} 1, & t = t_m, \\ 0, & \text{elsewhere}. \end{cases} \tag{3.10}$$

We select the following test functions

$$v_h^k \varphi^k(t), \ \forall k = 0, \ldots, m, \tag{3.11}$$

where $v_h^k \in V_h^k, \ \forall k = 0, \ldots, m$ and $\varphi^k(t)$ are the functions defined in (2.13).

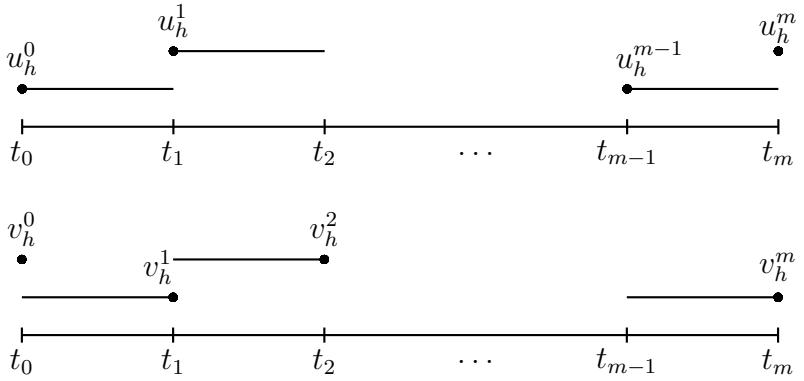



Figure 3.2.: Trial and test functions for the dPG formulation ($q = r = 0$).

*Theorem* 3.1. (Forward Euler method) Selecting trial and test functions as in (3.9) and (3.11), problem (3.6) leads to the following scheme

$$\begin{cases} \text{Find } u_h^0 \in V_h^0 \text{ and } u_h^k \in V_h^k, \ \forall k = 1, \ldots, m, \text{ such that} \\ \\ \left(u_h^k, v_h^k\right) = \left(u_h^{k-1}, v_h^k\right) - \tau_k \left(\nabla u_h^{k-1}, \nabla v_h^k\right) + \displaystyle\int_{I_k} \left\langle f, v_h^k \right\rangle dt, \ \forall v_h^k \in V_h^k, \\ \\ \left(u_h^0, v_h^0\right) = \left(u_0, v_h^0\right), \ \forall v_h^0 \in V_h^0, \end{cases} \tag{3.12}$$

which is an explicit method that is a variant of the Forward Euler method in time.

*Proof.* From the first test function of (3.11), $v_{\tau h}(t) = v_h^0 \varphi^0(t)$, and substituting it and the solution (3.9) on (3.6), we obtain

$$(u_h^0, v_h^0) = (u_0, v_h^0), \ \forall v_h^0 \in V_h^0.$$

Then, for $k = 1, \ldots, m$, each test function of (3.11) has local support in $(t_{k-1}, t_k]$, so problem (3.6) reads

$$\int_{I_k} \langle u_{\tau h,t}, v_{\tau h} \rangle \, dt + \int_{I_k} (\nabla u_{\tau h}, \nabla v_{\tau h}) dt + (\llbracket u_{\tau h} \rrbracket^k, v_{\tau h}(t_k^-)) = \int_{I_k} \langle f, v_{\tau h} \rangle \, dt.$$

Now, as both trial and test functions are piecewise constant in time, in the previous equation we obtain

$$\int_{I_k} \langle u_{\tau h,t}, v_{\tau h} \rangle \, dt = (u_h^{k-1}, v_h^k) \int_{I_k} \left( \frac{\partial}{\partial t} \phi^{k-1} \right) \varphi^k dt = 0,$$

$$\int_{I_k} (\nabla u_{\tau h}, \nabla v_{\tau h}) dt = (\nabla u_h^{k-1}, \nabla v_h^k) \int_{I_k} \phi^{k-1} \varphi^k dt = \tau_k (\nabla u_h^{k-1}, \nabla v_h^k),$$

$$(\llbracket u_{\tau h} \rrbracket^k, v_{\tau h}(t_k^-)) = (u_h^k \phi^k(t_k^+) - u_h^{k-1} \phi^{k-1}(t_k^-), v_h^k \varphi^k(t_k^-)) = (u_h^k - u_h^{k-1}, v_h^k),$$

$$\int_{I_k} \langle f, v_{\tau h} \rangle \, dt = \int_{I_k} \left\langle f, v_h^k \right\rangle dt.$$

Finally, we obtain

$$\left(u_h^k, v_h^k\right) = \left(u_h^{k-1}, v_h^k\right) - \tau_k \left(\nabla u_h^{k-1}, \nabla v_h^k\right) + \int_{I_k} \left\langle f, v_h^k \right\rangle dt.$$

□

Scheme (3.12) is the Forward Euler method in time except for the source term. A standard difference between variational forms and difference methods is that variational forms include an integral measure rather than a point-wise sample of the forcing terms. To obtain an expression whose form is identical to the classical Forward Euler method, we can interpolate the source term as

$$f(\mathbf{x}, t) = \sum_{k=1}^{m+1} f^{k-1}(\mathbf{x}) \phi^{k-1}(t),$$

where we identify $f^k(\mathbf{x})$ with $f(\mathbf{x}, t_k)$. Then, the source term in (3.12) becomes

$$\int_{I_k} \left\langle f, v_h^k \right\rangle dt = \int_{I_k} \left\langle f^{k-1} \phi^{k-1}, v_h^k \right\rangle dt = \tau_k \left\langle f^{k-1}, v_h^k \right\rangle.$$

In space, we can then employ a Spectral Element Method (SEM), which leads to a diagonal mass matrix for arbitrary dimensional problems using arbitrary geometrical mappings [119].

Finally, as (3.12) is an explicit method, it is conditionally stable, which implies that the time step size should be constrained by the spatial resolution to keep the method stable. Thus, we ensure that the Courant-Friedrichs-Lewy (CFL) condition [132] is satisfied to recover proper stability of the method (See Appendix B.1 for details).

*Remark* 3.1. In the subspaces defined in (3.8), as $v_h^k \in V_h^k,\ \forall k = 0, \dots, m$, the spatial discrete spaces in the test space $\tilde{\mathcal{V}}_{\tau h}$ are displaced in time with respect to the trial space $\tilde{\mathcal{U}}_{\tau h}$, which leads to a Petrov-Galerkin method. This is required to obtain invertible square mass matrices on the left-hand-side of (3.12). Figure 3.3 illustrates this displacement of the spaces.

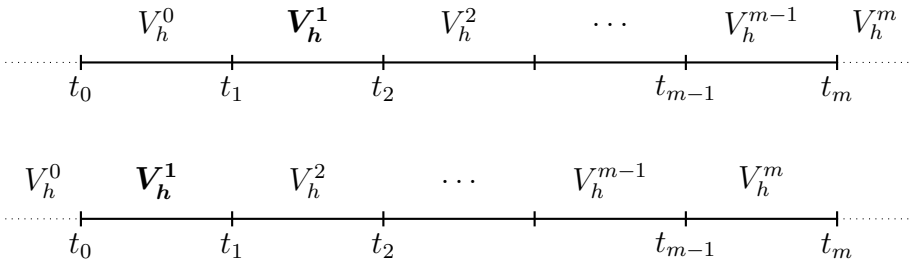



Figure 3.3.: Illustration of the displacement in time of the trial (top) and test (bottom) discrete spaces for the lowest order case ($q = r = 0$).

## 3.2.2. Two-stage RK methods

For two-stage RK methods, we introduce a test function for the initial condition $v_h\varphi^0(t)$ with $v_h \in V_h^0$ and being $\varphi^0(t)$ the function defined in (2.13). We also introduce two test functions per temporal element $I_k$

$$v_{h,1}\varphi_1^k(t),\ v_{h,2}\varphi_2^k(t), \tag{3.13}$$

where $v_{h,1} \in V_h^k,\ v_{h,2} \in V_h^{k-1},\ \forall k = 1, \dots, m$. Then, we express the solution of (3.6) as

$$u_{\tau h}(t) = \sum_{k=1}^{m} u_{h,1}^{k-1}\phi_1^{k-1}(t) + u_{h,2}^{k-1}\phi_2^{k-1}(t) + u_h^m\phi^m(t), \tag{3.14}$$

where $u_{h,1}^{k-1}, u_{h,2}^{k-1} \in V_h^{k-1},\ \forall k = 1, \dots, m,\ u_h^m \in V_h^m$ and $\phi^m(t)$ is the function defined in (3.10).

As before, the trial and test functions are piecewise polynomials defined in $I_k$ and globally discontinuous across the time interfaces. We define the subspaces in (3.6) following a similar construction as in [128]

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h} &:= \{u \in L^2(I;V) \,|\, u_{|_{[t_{k-1},t_k)}} \in \tilde{\mathcal{U}}_{\tau h}^{k-1},\ \forall k = 1, \dots, m,\ u(T) \in V_h^m\}, \\ \tilde{\mathcal{V}}_{\tau h} &:= \{v \in L^2(I;V) \,|\, v_{|_{(t_{k-1},t_k]}} \in \tilde{\mathcal{V}}_{\tau h}^{k},\ \forall k = 1, \dots, m,\ v(0) \in V_h^0\}, \end{aligned} \tag{3.15}$$

where we define the following discrete subspaces in each interval

$$\begin{aligned} \tilde{\mathcal{U}}_{\tau h}^{k-1} &:= \operatorname{span}\left\{u_{h,1}^{k-1}\phi_1^{k-1}(t),\ u_{h,2}^{k-1}\phi_2^{k-1}(t) \mid u_{h,1}^{k-1},\ u_{h,2}^{k-1} \in V_h^{k-1}\right\}, \\ \tilde{\mathcal{V}}_{\tau h}^{k} &:= \operatorname{span}\left\{v_{h,1}\varphi_1^k(t),\ v_{h,2}\varphi_2^k(t) \mid v_{h,1} \in V_h^k,\ v_{h,2} \in V_h^{k-1}\right\}. \end{aligned}$$

Notice that $\tilde{\mathcal{U}}_{\tau h}^{k-1} \subset P_q^{inc_u}([t_{k-1}, t_k); V)$ and $\tilde{\mathcal{V}}_{\tau h}^{k} \subset P_r^{inc_v}((t_{k-1}, t_k]; V)$ and therefore, $\tilde{\mathcal{U}}_{\tau h} \subset \tilde{\mathcal{U}}_{\tau}$ and $\tilde{\mathcal{V}}_{\tau h} \subset \tilde{\mathcal{V}}_{\tau}$.

We build the trial and test functions that generate the spaces (3.15) to obtain equivalent methods to some explicit two-stage and second-order RK methods.

*Remark* 3.2. In order for the discrete system to make sense and obtain invertible square mass matrices at each stage, we need the test functions (3.13) to satisfy

$$v_{h,1} \in V_h^k, \quad v_{h,2} \in V_h^{k-1},$$

so while both test functions are polynomials in time, in space they belong to different spaces. Figure 3.4 illustrates this choice of the test functions.

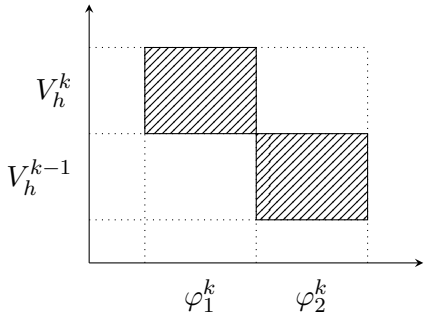



Figure 3.4.: Space-time structure of the test functions inside each element $I_k$.

We assume that each trial function is associated with a coefficient (as in Figure 3.5). Thus, we impose

$$\begin{cases} \phi_1^{k-1}(t_{k-1}^+) = 1, \\ \phi_2^{k-1}(t_{k-1}^+) = 0, \end{cases} \tag{3.16}$$

so $u_{h,1}^{k-1} \in V_h^{k-1}$ is the value of $u_{\tau h}(t)$ at $t_{k-1}^+$.

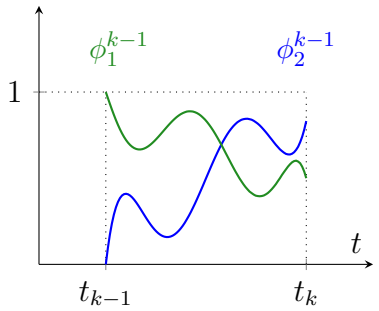



Figure 3.5.: Trial functions of arbitrary order inside each element $I_k$.

We first substitute (3.14) and the first test function $v_h\varphi^0(t)$ into (3.6) and we obtain

$$(u_{h,1}^0, v_h) = (u_0, v_h),\ \forall v_h \in V_h^0.$$

Now, if we substitute (3.14) into (3.6), as each test function in (3.13) has local support in $I_k$, we have

$$B_{DG}(u_{h,1}^{k-1}\phi_1^{k-1}, v_{h,j}\varphi_j^k) + B_{DG}(u_{h,2}^{k-1}\phi_2^{k-1}, v_{h,j}\varphi_j^k) = F(v_{h,j}\varphi_j^k), \tag{3.17}$$

where $j \in \{1, 2\}$. In the above equation, the jump of $u_{\tau h}(t)$ at $t_k$ is

$$\begin{aligned}\llbracket u_{\tau h}\rrbracket^k &= u_{\tau h}(t_k^+) - u_{\tau h}(t_k^-)\\ &= u_{h,1}^k\phi_1^k(t_k^+) + u_{h,2}^k\phi_2^k(t_k^+) - u_{h,1}^{k-1}\phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1}\phi_2^{k-1}(t_k^-)\\ &= u_{h,1}^k - u_{h,1}^{k-1}\phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1}\phi_2^{k-1}(t_k^-),\end{aligned}$$

$\forall k = 1, \ldots, m-1$ and, in particular, for the last time step, we have

$$\begin{aligned}\llbracket u_{\tau h}\rrbracket^m &= u_{\tau h}(t_m^+) - u_{\tau h}(t_m^-)\\ &= u_h^m\phi^m(t_m^+) - u_{h,1}^{m-1}\phi_1^{m-1}(t_m^-) - u_{h,2}^{m-1}\phi_2^{m-1}(t_m^-)\\ &= u_h^m - u_{h,1}^{m-1}\phi_1^{m-1}(t_m^-) - u_{h,2}^{m-1}\phi_2^{m-1}(t_m^-).\end{aligned}$$

Equivalently, we write equation (3.17) as

$$\begin{aligned}&\left(u_{h,1}^{k-1}, v_{h,j}\right)\int_{I_k}\left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_j^k dt + \left(u_{h,2}^{k-1}, v_{h,j}\right)\int_{I_k}\left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_j^k dt\\ &+ \left(\nabla u_{h,1}^{k-1}, \nabla v_{h,j}\right)\int_{I_k}\phi_1^{k-1}\varphi_j^k dt + \left(\nabla u_{h,2}^{k-1}, \nabla v_{h,j}\right)\int_{I_k}\phi_2^{k-1}\varphi_j^k dt\\ &+ \left(u_{h,1}^k - u_{h,1}^{k-1}\phi_1^{k-1}(t_k^-) - u_{h,2}^{k-1}\phi_2^{k-1}(t_k^-), v_{h,j}\varphi_j^k(t_k^-)\right) = \int_{I_k}\left\langle f, v_{h,j}\varphi_j^k\right\rangle dt.\end{aligned} \tag{3.18}$$

Now, reorganizing the terms in (3.18) we obtain

$$\begin{aligned}&\left(u_{h,1}^{k-1}, v_{h,j}\right)\left(\int_{I_k}\left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_j^k dt - \phi_1^{k-1}(t_k^-)\varphi_j^k(t_k^-)\right)\\ &+ \left(u_{h,2}^{k-1}, v_{h,j}\right)\left(\int_{I_k}\left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_j^k dt - \phi_2^{k-1}(t_k^-)\varphi_j^k(t_k^-)\right)\\ &+ \left(\nabla u_{h,1}^{k-1}, \nabla v_{h,j}\right)\int_{I_k}\phi_1^{k-1}\varphi_j^k dt\\ &+ \left(\nabla u_{h,2}^{k-1}, \nabla v_{h,j}\right)\int_{I_k}\phi_2^{k-1}\varphi_j^k dt\\ &+ (u_{h,1}^k, v_{h,j})\varphi_j^k(t_k^-) = \int_{I_k}\left\langle f, v_{h,j}\varphi_j^k\right\rangle dt.\end{aligned} \tag{3.19}$$

We build the trial and test functions to guarantee the satisfaction of some design conditions. We need the following orthogonality conditions in order to obtain an explicit method

$$\begin{cases} \varphi_2^k(t_k^-) = 0, \\ \displaystyle\int_{I_k} \phi_2^{k-1}\varphi_2^k dt = 0. \end{cases} \tag{3.20}$$

In (3.19) we have two equations (one per test function) and $u_{h,1}^{k-1}$ is a known value from the previous step. By imposing (3.20), we can calculate explicitly $u_{h,2}^{k-1}$ from $u_{h,1}^{k-1}$ and then compute $u_h^k$ from $u_{h,1}^{k-1}$ and $u_{h,2}^{k-1}$. To obtain a RK method, we need to impose further conditions on the system. Indeed, the general expression of the two-stage and second-order explicit RK method we want to obtain is

$$\begin{aligned} (u_{h,1}^k, v_{h,1}) \; -(u_{h,1}^{k-1}, v_{h,1}) + \frac{\tau_k}{2\alpha}(\nabla u_{h,2}^{k-1}, \nabla v_{h,1}) & \\ + \quad \left(1 - \frac{1}{2\alpha}\right)\tau_k(\nabla u_{h,1}^{k-1}, \nabla v_{h,1}) \quad &= \int_{I_k} \left\langle f, v_{h,1}\varphi_1^k \right\rangle dt, \\ (u_{h,2}^{k-1}, v_{h,2}) - (u_{h,1}^{k-1}, v_{h,2}) + \alpha\tau_k(\nabla u_{h,1}^{k-1}, \nabla v_{h,2}) &= \int_{I_k} \left\langle f, v_{h,2}\varphi_2^k \right\rangle dt, \end{aligned} \tag{3.21}$$

where $\alpha \in \mathbb{R} - \{0\}$ and its corresponding Butcher tableau is described in Table 3.1.

| | | |
|---|---|---|
| 0 | 0 | 0 |
| $\alpha$ | $\alpha$ | 0 |
| | $1 - \frac{1}{2\alpha}$ | $\frac{1}{2\alpha}$ |

Table 3.1.: Butcher tableau for two-stage and second-order explicit RK methods.

In order to obtain (3.21) from (3.19), in addition to the orthogonality conditions (3.20), we need to impose also the following conditions from the first equation of (3.21):

$$\begin{cases} \varphi_1^k(t_k^-) = 1, \\ \displaystyle\int_{I_k} \left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_1^k dt - \phi_2^{k-1}(t_k^-)\varphi_1^k(t_k^-) = 0, \\ \displaystyle\int_{I_k} \left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_1^k dt - \phi_1^{k-1}(t_k^-)\varphi_1^k(t_k^-) = -1, \\ \displaystyle\int_{I_k} \phi_2^{k-1}\varphi_1^k dt = \frac{\tau_k}{2\alpha}, \\ \displaystyle\int_{I_k} \phi_1^{k-1}\varphi_1^k dt = \left(1 - \frac{1}{2\alpha}\right)\tau_k, \end{cases} \tag{3.22}$$

and also from the second equation of (3.21):

$$\begin{cases} \displaystyle\int_{I_k} \left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_2^k dt - \phi_2^{k-1}(t_k^-)\varphi_2^k(t_k^-) = 1, \\ \displaystyle\int_{I_k} \left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_2^k dt - \phi_1^{k-1}(t_k^-)\varphi_2^k(t_k^-) = -1, \\ \displaystyle\int_{I_k} \phi_1^{k-1}\varphi_2^k dt = \alpha\tau_k. \end{cases} \tag{3.23}$$

Finally, from (3.16), (3.20), (3.22) and (3.23), the conditions that the trial and test functions must satisfy to obtain the general two-stage explicit RK method (3.21) are

$$\begin{aligned} &\varphi_1^k(t_k^-) = 1, && \varphi_2^k(t_k^-) = 0, \\ &\phi_1^{k-1}(t_{k-1}^+) = 1, && \phi_2^{k-1}(t_{k-1}^+) = 0, \\ &\int_{I_k} \left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_1^k dt - \phi_1^{k-1}(t_k^-) = -1, && \int_{I_k} \left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_1^k dt - \phi_2^{k-1}(t_k^-) = 0, \\ &\int_{I_k} \left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_2^k dt = -1, && \int_{I_k} \left(\frac{\partial}{\partial t}\phi_2^{k-1}\right)\varphi_2^k dt = 1, \\ &\int_{I_k} \phi_1^{k-1}\varphi_1^k dt = \left(1 - \frac{1}{2\alpha}\right)\tau_k, && \int_{I_k} \phi_2^{k-1}\varphi_1^k dt = \frac{\tau_k}{2\alpha}, \\ &\int_{I_k} \phi_1^{k-1}\varphi_2^k dt = \alpha\tau_k, && \int_{I_k} \phi_2^{k-1}\varphi_2^k dt = 0. \end{aligned} \tag{3.24}$$

For the trial and test functions, which are polynomials of arbitrary order, conditions (3.24) become a system of nonlinear equations. In particular, if we select linear trial and test functions, we obtain no solutions. However, if we select quadratic functions, we have a system of 12 nonlinear equations with 12 unknowns, which has two solutions. We solve the resulting system in the master element $[0, 1]$ using the MATLAB code we describe in Appendix A.2 (see Table 3.2), and we obtain different sets of trial and test functions depending on the value of $\alpha$.

As in Section 3.2.1, to obtain expressions whose form is identical to standard RK methods, we can interpolate the source term in the trial space as

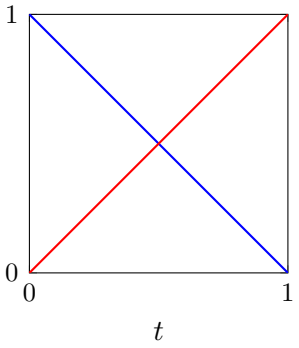
$$\begin{aligned} \int_{I_k} \left\langle f, v_{h,1}\varphi_1^k \right\rangle dt &= \int_{I_k} \left\langle f_1^{k-1}\phi_1^{k-1} + f_2^{k-1}\phi_2^{k-1}, v_{h,1}\varphi_1^k \right\rangle dt \\ &= \left(1 - \frac{1}{2\alpha}\right)\tau_k \left\langle f_1^{k-1}, v_{h,1} \right\rangle + \frac{\tau_k}{2\alpha} \left\langle f_2^{k-1}, v_{h,1} \right\rangle, \\ \int_{I_k} \left\langle f, v_{h,2}\varphi_2^k \right\rangle dt &= \int_{I_k} \left\langle f_1^{k-1}\phi_1^{k-1} + f_2^{k-1}\phi_2^{k-1}, v_{h,2}\varphi_2^k \right\rangle dt = \alpha\tau_k \left\langle f_1^{k-1}, v_{h,2} \right\rangle, \end{aligned}$$

which is the general two-stage RK method in time. Here, we identify $f_1^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1})$ and $f_2^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1} + \alpha\tau_k)$.

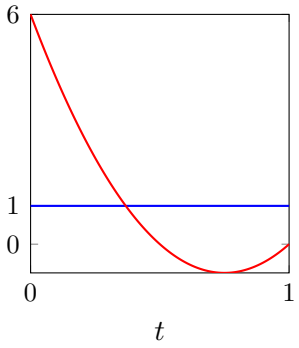
| | Solution 1 | Solution 2 |
|---|---|---|
| $\phi_1^{k-1}(t)$ | $-\frac{1}{\alpha}t + 1$ | $-\frac{6}{\alpha}t^2 + \frac{3}{\alpha}t + 1$ |
| $\phi_2^{k-1}(t)$ | $\frac{1}{\alpha}t$ | $\frac{6}{\alpha}t^2 - \frac{3}{\alpha}t$ |
| $\varphi_1^k(t)$ | $1$ | $1$ |
| $\varphi_2^k(t)$ | $12\alpha t^2 - 18\alpha t + 6\alpha$ | $-2\alpha t + 2\alpha$ |

Table 3.2.: Trial and test functions defined over the master element $[0, 1]$ that lead to the two-stage explicit RK method.

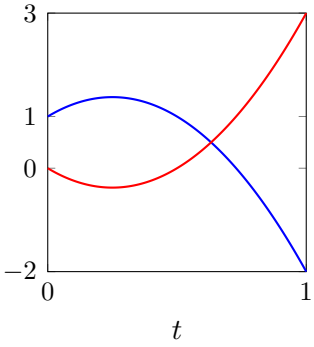
*Example* 3.1. (Explicit trapezoidal rule) When $\alpha = 1$, (3.21) is equivalent to the explicit trapezoidal rule (or Heun's method) [75]. Figure 3.6 shows the trial and test functions of both solutions over the master element.



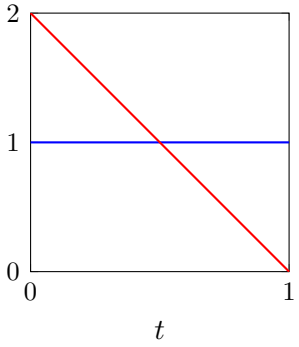

(a) Trial functions of the first solution.




(b) Test functions of the first solution.



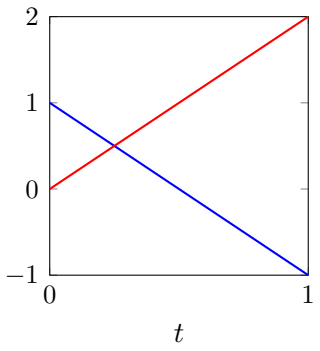

(c) Trial functions of the second solution.



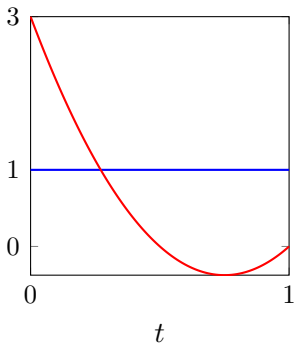

(d) Test functions of the second solution.

Figure 3.6.: Trial and test functions over the master element $[0, 1]$ when $\alpha = 1$.

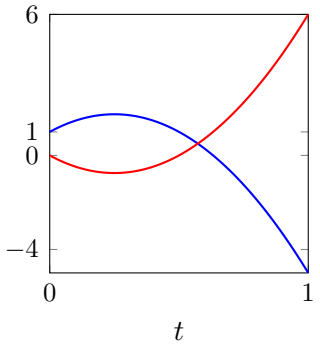
*Example* 3.2. (Explicit midpoint rule) When $\alpha = \frac{1}{2}$, we obtain the explicit midpoint rule [75]. Figure 3.7 shows the trial and test functions of both solutions over $[0, 1]$.



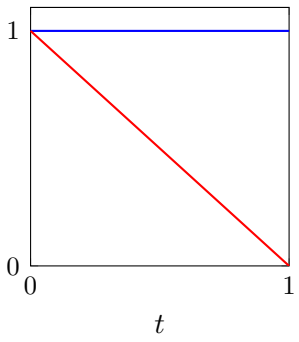

(a) Trial functions of the first solution.




(b) Test functions of the first solution.




(c) Trial functions of the second solution.




(d) Test functions of the second solution.

Figure 3.7.: Trial and test functions over the master element $[0, 1]$ when $\alpha = \frac{1}{2}$.

In these examples, we set $q = r = 2$ in (3.3) and we have two possibilities for the subspaces in problem (3.6). In the first solution of Table 3.2, the trial space $\tilde{\mathcal{U}}_{\tau h}$ is the complete space of linear polynomials, while in the second solution it is an incomplete space of quadratic polynomials that only reproduces constant functions in time. Both solutions are nodally equivalent to the same two-stage and second-order explicit RK method. We will focus on the first solution in Table 3.2 so we have the mathematical theory of approximability inside the temporal intervals.

### 3.2.3. General s-stage explicit RK Methods

The general method we want to obtain is of the form

$$\begin{aligned}(u_{h,1}^k, v_{h,1}) \; - (u_{h,1}^{k-1}, v_{h,1}) + \tau_k \sum_{i=1}^{s} b_i (\nabla u_{h,i}^{k-1}, \nabla v_{h,1}) \; &= \tau_k \sum_{i=1}^{s} b_i \left\langle f_i^{k-1}, v_{h,1} \right\rangle, \\ (u_{h,i}^{k-1}, v_{h,i}) - (u_{h,1}^{k-1}, v_{h,i}) \; + \tau_k \sum_{j=1}^{i-1} a_{ij} (\nabla u_{h,j}^{k-1}, \nabla v_{h,i}) &= \tau_k \sum_{j=1}^{i-1} a_{ij} \left\langle f_j^{k-1}, v_{h,i} \right\rangle,\end{aligned} \tag{3.25}$$

$\forall i = 2, \ldots, s$, where

$$f_i^{k-1}(\mathbf{x}) := f(\mathbf{x}, t_{k-1} + c_i \tau_k), \; \forall i = 1, \ldots, s.$$

The coefficients $a_{ij}$, $b_i$, $c_i$, with $i, j \in \{1, \ldots, s\}$, are the ones corresponding to the Butcher tableau (see Table 3.3) [26]. As (3.25) is an explicit method, we have that

$$a_{ij} = 0, \forall j \geqslant i.$$

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ... | 0 |
| $c_2$ | $a_{21}$ | 0 | 0 | ... | 0 |
| $c_3$ | $a_{31}$ | $a_{32}$ | 0 | ... | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\ddots$ | $\vdots$ |
| $c_s$ | $a_{s1}$ | $a_{s2}$ | ... | $a_{s,s-1}$ | 0 |
| | $b_1$ | $b_2$ | ... | $b_{s-1}$ | $b_s$ |

Table 3.3.: Butcher tableau for the general s-stage explicit RK method.

We consider $s$ trial functions per time interval

$$\phi_1^{k-1}(t), \ldots, \phi_s^{k-1}(t),$$

so we express the solution in (3.6) as

$$u_{\tau h}(t) = \sum_{k=1}^{m} \sum_{i=1}^{s} u_{h,i}^{k-1} \phi_i^{k-1}(t) + u_h^m \phi^m(t).$$

We also consider $s$ test functions per time interval

$$v_{h,1}\varphi_1^k(t), \; v_{h,2}\varphi_2^k(t), \ldots, v_{h,s}\varphi_s^k(t),$$

$\forall k = 1, \ldots, m$ and we define the subspaces in (3.6) as

$$\tilde{\mathcal{U}}_{\tau h} := \{u \in L^2(I;V) \,|\, u_{|_{[t_{k-1},t_k)}} \in \tilde{\mathcal{U}}_{\tau h}^{k-1},\ \forall k = 1,\dots,m,\ u(T) \in V_h^m\},$$
$$\tilde{\mathcal{V}}_{\tau h} := \{v \in L^2(I;V) \,|\, v_{|_{(t_{k-1},t_k]}} \in \tilde{\mathcal{V}}_{\tau h}^{k},\ \forall k = 1,\dots,m,\ v(0) \in V_h^0\},$$

where we define

$$\tilde{\mathcal{U}}_{\tau h}^{k-1} := \operatorname{span}\left\{u_{h,i}^{k-1}\phi_i^{k-1}(t) \mid u_{h,i}^{k-1} \in V_h^{k-1},\ \forall i = 1,\dots,s\right\},$$
$$\tilde{\mathcal{V}}_{\tau h}^{k} := \operatorname{span}\left\{v_{h,1}\varphi_1^k(t),\ v_{h,j}\varphi_j^k(t) \mid v_{h,1} \in V_h^k,\ v_{h,j} \in V_h^{k-1},\ \forall j = 2,\dots,s\right\}.$$

*Remark* 3.3. Again, to properly define solvable discrete systems, we seek to obtain square mass matrices in (3.25). Thus, we need test functions satisfying

$$v_{h,1} \in V_h^k,\ \ v_{h,j} \in V_h^{k-1},\ \forall j = 2,\dots,s.$$

Figure 3.8 illustrates this choice of the test functions.

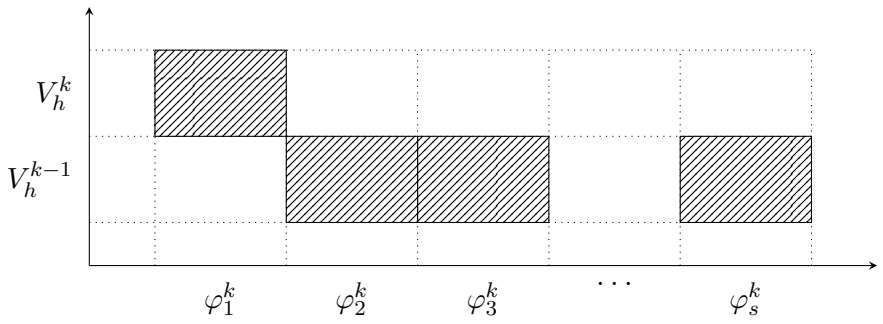



Figure 3.8.: Space-time structure of the test functions inside each element $I_k$.

As before, to obtain an expression whose form is identical to the classical RK methods, we can interpolate the source term using the trial functions as

$$f(\mathbf{x},t) = \sum_{k=1}^{m+1}\sum_{i=1}^{s} f_i^{k-1}(\mathbf{x})\phi_i^{k-1}(t).$$

Following the same logic as in Section 3.2.2, we generalize conditions (3.24) to $s$-stages as follows

$$
\begin{aligned}
&\varphi_1^k(t_k^-) = 1, && \varphi_j^k(t_k^-) = 0,\ \forall j = 2,\dots,s,\\
&\phi_1^{k-1}(t_{k-1}^+) = 1, && \phi_i^{k-1}(t_{k-1}^+) = 0,\ \forall i = 2,\dots,s,\\
&\int_{I_k}\left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_1^k dt - \phi_1^{k-1}(t_k^-) = -1, && \int_{I_k}\left(\frac{\partial}{\partial t}\phi_i^{k-1}\right)\varphi_1^k dt - \phi_i^{k-1}(t_k^-) = 0,\\
& && \forall i = 2,\dots,s,\\
&\int_{I_k}\left(\frac{\partial}{\partial t}\phi_1^{k-1}\right)\varphi_j^k dt = -1, && \int_{I_k}\left(\frac{\partial}{\partial t}\phi_i^{k-1}\right)\varphi_j^k dt = \begin{cases}1, \text{ if } i = j\\ 0, \text{ if } i \neq j\end{cases},\\
& && \forall i,j = 2,\dots,s,\\
&\int_{I_k}\phi_i^{k-1}\varphi_1^k dt = \tau_k b_i,\ \forall i = 1,\dots,s, && \\
&\int_{I_k}\phi_i^{k-1}\varphi_j^k dt = \tau_k a_{ji},\ \forall i = 1,\dots,s, && \forall j = 2,\dots,s.
\end{aligned}
\tag{3.26}
$$

Appendix A.1 expresses (3.26) in matrix form.

*Remark* 3.4. We can extend this construction to Ordinary Differential Equations (ODEs) of the form $\dot{u} = f(t,u(t))$, $u(0) = u_0$ by interpolating the right-hand-side as follows

$$f(t,u(t)) \simeq \sum_{k=1}^{m+1}\sum_{i=1}^{s} f(t_{k-1} + \tau_k c_i, u_{h,i}^{k-1})\phi_i^{k-1}(t)$$

where $\text{span}\{\phi_i^{k-1}\}_{i=1}^s$ is a complete space of polynomials.

*Example* 3.3. (Three-stage RK) We calculate the trial and test functions of the three-stage and third-order explicit RK method with Butcher tableau defined in Table 3.4 [26].

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 0 |
| 1 | $-1$ | 2 | 0 |
| | $\frac{1}{6}$ | $\frac{2}{3}$ | $\frac{1}{6}$ |

Table 3.4.: Butcher tableau of a three-stage and third-order explicit RK method.

If we consider cubic polynomials ($q = r = 3$) in (3.3), we obtain four possible solutions for the trial and test basis functions in the subspaces (3.6): two of them with real coefficients (see Table 3.5) and the remaining two solutions with complex conjugate coefficients (see Table 3.6). In Table 3.6, $z_j$ and $\bar{z}_j$ denote the following complex numbers and their conjugates.

$$
\begin{aligned}
z_1 &= (9 - i\sqrt{66})/7, & z_6 &= (34 + 3i\sqrt{66})/7,\\
z_2 &= (11 - 2i\sqrt{66})/7, & z_7 &= (16 + i\sqrt{66})/7,\\
z_3 &= (12 + i\sqrt{66})/7, & z_8 &= (30 + i\sqrt{66})/7,\\
z_4 &= (5 + i\sqrt{66})/7, & z_9 &= (89 + 6i\sqrt{66})/7,\\
z_5 &= (2 - i\sqrt{66})/7, & z_{10} &= (39 + 2i\sqrt{66})/7,
\end{aligned}
$$

| | Solution 1 | Solution 2 |
|---|---|---|
| $\phi_1^{k-1}(t)$ | $\frac{1}{2}t^2 - 2t + 1$ | $110t^3 - 130t^2 + 30t + 1$ |
| $\phi_2^{k-1}(t)$ | $-t^2 + 2t$ | $-120t^3 + 140t^2 - 32t$ |
| $\phi_3^{k-1}(t)$ | $\frac{1}{2}t^2$ | $10t^3 - 10t^2 + 2t$ |
| $\varphi_1^k(t)$ | $1$ | $1$ |
| $\varphi_2^k(t)$ | $-30t^3 + 60t^2 - 36t + 6$ | $\frac{3}{2}t^2 - 3t + \frac{3}{2}$ |
| $\varphi_3^k(t)$ | $420t^3 - 780t^2 + 408t - 48$ | $-6t^2 + 6t$ |

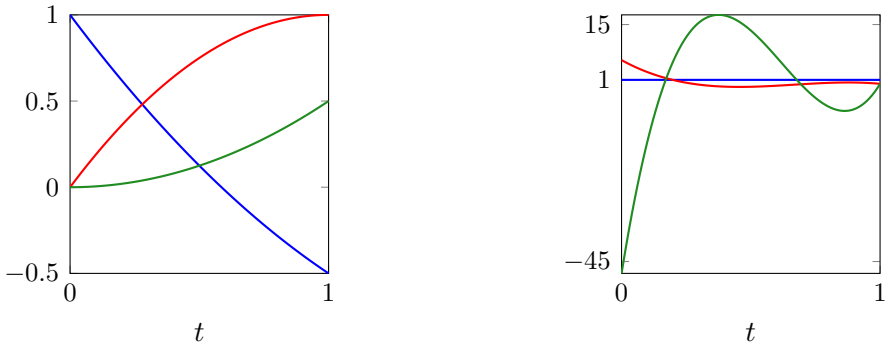
Table 3.5.: Trial and test functions with real coefficients over the master element $[0, 1]$ that lead to a three-stage explicit RK method.

| | Solution 3 | Solution 4 |
|---|---|---|
| $\phi_1^{k-1}(t)$ | $\frac{10}{3}z_1t^3 - 2z_2t^2 - z_3t + 1$ | $\frac{10}{3}\bar{z}_1t^3 - 2\bar{z}_2t^2 - \bar{z}_3t + 1$ |
| $\phi_2^{k-1}(t)$ | $-\frac{20}{3}z_1t^3 + 4z_2t^2 + 2z_4t$ | $-\frac{20}{3}\bar{z}_1t^3 + 4\bar{z}_2t^2 + 2\bar{z}_4t$ |
| $\phi_3^{k-1}(t)$ | $\frac{10}{3}z_1t^3 - 2z_2t^2 + z_5t$ | $\frac{10}{3}\bar{z}_1t^3 - 2\bar{z}_2t^2 + \bar{z}_5t$ |
| $\varphi_1^k(t)$ | $1$ | $1$ |
| $\varphi_2^k(t)$ | $-10\bar{z}_1t^3 + 6z_6t^2 - 9z_7t + z_8$ | $-10z_1t^3 + 6\bar{z}_6t^2 - 9\bar{z}_7t + \bar{z}_8$ |
| $\varphi_3^k(t)$ | $40\bar{z}_1t^3 - 24z_6t^2 + 6z_9t - 2z_{10}$ | $40z_1t^3 - 24\bar{z}_6t^2 + 6\bar{z}_9t - 2\bar{z}_{10}$ |

Table 3.6.: Trial and test functions with complex coefficients over the master element $[0, 1]$ that lead to a three-stage explicit RK method.

Here, the four solutions we obtain are nodally equivalent to the three-stage and third-order explicit RK method defined in Table 3.4. However, the utility of the complex solutions is unknown for the authors and an open area of research.

Figure 3.9 shows the trial and test functions of the first real solution over the master element $[0, 1]$. In this case, the trial space $\tilde{\mathcal{U}}_{\tau h}$ is a complete space of quadratic polynomials.




(a) Trial functions of the first real solution. (b) Test functions of the first real solution.

Figure 3.9.: Trial and test functions over the master element $[0, 1]$ for the real solutions of a three-stage explicit RK method.

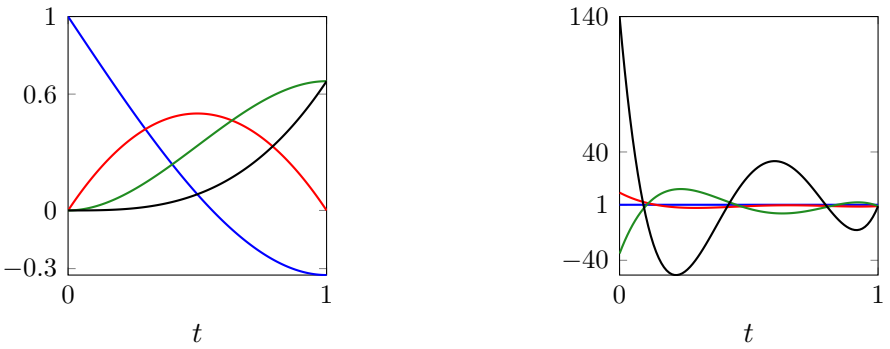
*Example* 3.4. (Four-stage RK) Now, we consider the four-stage and fourth-order explicit RK method with the Butcher tableau as Table 3.7 [26]. In order to reduce the number of unknowns, and taking into account the solutions we obtained for one and two stages, we solve the nonlinear system considering non-symmetric choices for the order of polynomials in (3.3): $q = 3$, $r = 4$ and $q = 4$, $r = 3$. Table 3.8 shows two real solutions for this method and Figure 3.10 shows the solution for which the trial space $\tilde{\mathcal{U}}_{\tau h}$ is a complete space of cubic polynomials.

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 0 | 0 |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| | $\frac{1}{6}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{6}$ |

Table 3.7.: Butcher tableau of a four-stage and fourth-order explicit RK method.

| | Solution 1 | Solution 2 |
|---|---|---|
| $\phi_1^{k-1}(t)$ | $\frac{2}{3}t^3 - 2t + 1$ | $-\frac{2800}{3}t^4 + 1610t^3 - 810t^2 + \frac{320}{3}t + 1$ |
| $\phi_2^{k-1}(t)$ | $-2t^2 + 2t$ | $\frac{4480}{3}t^4 - 2520t^3 + 1230t^2 - \frac{470}{3}t$ |
| $\phi_3^{k-1}(t)$ | $-\frac{4}{3}t^3 + 2t^2$ | $-\frac{1820}{3}t^4 + 980t^3 - 450t^2 + \frac{160}{3}t$ |
| $\phi_4^{k-1}(t)$ | $\frac{2}{3}t^3$ | $\frac{140}{3}t^4 - 70t^3 + 30t^2 - \frac{10}{3}t$ |
| $\varphi_1^k(t)$ | 1 | 1 |
| $\varphi_2^k(t)$ | $140t^4 - 350t^3 + 300t^2 - 100t + 10$ | $-2t^3 + 6t^2 - 6t + 2$ |
| $\varphi_3^k(t)$ | $-910t^4 + 2170t^3 - 1725t^2 + 500t - 35$ | $2t^3 - 3t^2 + 1$ |
| $\varphi_4^k(t)$ | $4480t^4 - 10360t^3 + 7890t^2 - 2150t + 140$ | $-6t^2 + 6t$ |

Table 3.8.: Trial and test functions with real coefficients over the master element $[0, 1]$ that lead to a four-stage explicit RK method.




(a) Trial functions of the first real solution. (b) Test functions of the first real solution.

Figure 3.10.: Trial and test functions over the master element $[0, 1]$ for the real solutions of a four-stage explicit RK method.

In conclusion, we have proved that the explicit RK methods can be expressed as dPG methods. We systematically build trial and test functions that, after exact integration in time, lead to one, two, and general stage explicit Runge-Kutta methods. We employ this approach to represent the error in the QoI and to design an explicit-in-time goal-oriented adaptive strategy in the next chapter.

# 4. Explicit-in-time Goal-Oriented Adaptivity (GOA)

In this chapter, we first introduce basic ideas about error representation and GOA. We define the dual problem we employ to represent the error in the Quantity of Interest (QoI). Then, we derive the classical error representation for the discontinuous-in-time Galerkin (dG) formulation and we explain the classical goal-oriented adaptive strategies existing in the literature for time-dependent Partial Differential Equations (PDEs). Next, we provide an error representation formula for the discontinuous-in-time Petrov-Galerkin (dPG) formulation we proposed in Chapter 3 and we verify numerically by calculating the effectivity index that it can be employed for GOA.

We also design an explicit-in-time goal-oriented adaptive algorithm. First, we solve the primal problem (2.2) employing the Forward Euler scheme described in (3.1). Then, we define a dPG formulation for the dual problem and we show that by selecting piecewise constant functions in time, we obtain a scheme that is also algebraically equivalent to the forward Euler method but running backwards in time. Subsequently, we solve both problems running in different directions with explicit time-marching schemes. Finally, we describe a goal-oriented adaptive algorithm that performs local refinements in space and global refinements in time based on the Courant-Friedrichs-Lewy (CFL) condition.

## 4.1. Dual problem and error representation for the classical dG formulation

We consider a QoI given by a linear functional $L : \mathcal{U} \subset \mathcal{V} \longrightarrow \mathbb{R}$, which is called the *output functional* and quantifies a relevant feature of the solution. We consider linear output functionals of the form

$$L(v) = \int_I \langle g, v \rangle \, dt + (z_T, v(T)), \tag{4.1}$$

where $g \in \mathcal{V}'$ and $z_T \in L^2(\Omega)$ are given functions.

In order to represent the error in the QoI [3], we introduce the following *dual problem*

$$\begin{aligned} -\int_I \langle z_t, v\rangle\, dt + \int_I (\nu\nabla z, \nabla v)dt - \int_I (\beta\cdot\nabla z, v)dt = \int_I \langle g, v\rangle\, dt,\ \forall v\in\mathcal{V}, \\ (z(T),\hat{v}) = (z_T,\hat{v}),\ \forall \hat{v}\in L^2(\Omega), \end{aligned} \tag{4.2}$$

where $B^*(\cdot,\cdot)$ is the adjoint operator of the form $B(\cdot,\cdot)$

$$B^*(z,v) := -\int_I \langle z_t, v\rangle\, dt + \int_I (\nu\nabla z, \nabla v)dt - \int_I (\beta\cdot\nabla z, v)dt.$$

The dual problem provides us with information about how the error in every space-time point affects the error in the QoI.

For this particular case, the corresponding strong formulation of (4.2) is

$$\begin{cases} -z_t - \nabla\cdot(\nu\nabla z) - \beta\cdot\nabla z = g & \text{in } \Omega\times I, \\ z = 0 & \text{on } \partial\Omega\times I, \\ z(T) = z_T & \text{in } \Omega, \end{cases} \tag{4.3}$$

and we conclude that the dual problem runs backwards in time.

In the same way as the primal problem, we can solve the dual problem (4.2) employing time-marching scheme but running backwards in time. The dG formulation of problem (4.2) is

$$\begin{cases} \text{Find } z_{\tau h}\in\mathcal{V}_{\tau h}^r \text{ such that} \\ B_{DG}^{*+}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}),\ \forall v_{\tau h}\in\mathcal{V}_{\tau h}^r, \end{cases} \tag{4.4}$$

where $B_{DG}^{*+}(\cdot,\cdot)$ is the resulting bilinear form after integrating by parts in time and the advection term in space the form $B_{DG}^{+}(\cdot,\cdot)$.

Now, we proceed to derive the classical error representation [116] for the dG formulation we derived in Section 2.2.1. For the continuous-in-time Petrov-Galerkin (cPG) case see [9, 14].

First, notice that problems (2.2) holds for any subspace of $\mathcal{V}$. In particular, we select $\mathcal{V}_\tau^r$ and adding both equations in (2.2) respectively, we obtain

$$B(u, v_\tau) + (u(0), v_\tau(0^-)) = F(v_\tau),\ \forall v_\tau\in\mathcal{V}_\tau^r,$$

and as the solution of (2.2) is globally continuous in time, the jump terms at each time interface are zero so we have

$$B(u, v_\tau) + (u(0), v_\tau(0^-)) = B_{DG}^{+}(u, v_\tau),\ \forall v_\tau\in\mathcal{V}_\tau^r. \tag{4.5}$$

Therefore, we consider the following continuous and discrete primal problems

$$\begin{cases} \text{Find } u \in \mathcal{U} \text{ and } u_{\tau h} \in \mathcal{V}_{\tau h}^r \text{ such that} \\ B_{DG}^+(u, v_\tau) = F(v_\tau),\ \forall v_\tau \in \mathcal{V}_\tau^r, \\ B_{DG}^+(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}_{\tau h}^r. \end{cases} \tag{4.6}$$

Now, as $\mathcal{V}_\tau^r \nsubseteq \mathcal{U}$ (because functions in $\mathcal{V}_\tau^r$ are discontinuous) and $\mathcal{U} \nsubseteq \mathcal{V}_\tau^r$ (since functions in $\mathcal{V}_\tau^r$ are piecewise polynomials), we define the following space

$$\mathcal{V}_\tau = \{v \in \mathcal{V} \mid v_{|_{I_k}} \in L^2(I_k; V),\ v_{t_{|_{I_k}}} \in L^2(I_k; V'),\ \forall k = 1, \dots, m\}, \tag{4.7}$$

which is the minimum subspace of $\mathcal{V}$ containing both $\mathcal{V}_\tau^r$ and $\mathcal{U}$.

Similarly as for the primal problem, (4.2) holds for every subspace of $\mathcal{V}$. So selecting $\mathcal{V}_\tau \subset \mathcal{V}$, adding both equations of (4.2) and taking into account that the jumps of $z$ at each time interface are zero, we consider

$$\begin{cases} \text{Find } z \in \mathcal{U} \text{ such that} \\ B_{DG}^{*+}(z, v_\tau) = L(v_\tau),\ \forall v_\tau \in \mathcal{V}_\tau. \end{cases} \tag{4.8}$$

We define the error of the primal problem as $e = u - u_{\tau h} \in \mathcal{V}_\tau$ and we represent the error in the QoI as an integral over the whole space-time domain $\Omega \times I$ in the following theorem.

*Theorem* 4.1. (Error representation dG) Let $B_{DG}^+(\cdot,\cdot)$ be the bilinear form defined in Eq. (2.8), $L(\cdot)$ the linear output functional defined in Eq. (4.1) and $e = u - u_{\tau h}$ the error of the primal problem. It holds that

$$L(e) = B_{DG}^+(e, z - v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}_{\tau h}, \tag{4.9}$$

and alternatively

$$L(e) = F(z - v_{\tau h}) - B_{DG}^+(u_{\tau h}, z - v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}_{\tau h}, \tag{4.10}$$

*Proof.* We have that $u \in \mathcal{U} \subset \mathcal{V}_\tau$ and $u \in \mathcal{V}_{\tau h}^r \subset \mathcal{V}_\tau^r \subset \mathcal{V}_\tau$ so $e = u - u_{\tau h} \in \mathcal{V}_\tau$. Substituting $e$ into (4.8) and integrating by parts in time and in space the advection term, we obtain

$$L(e) = B_{DG}^{*+}(z, e) = B_{DG}^+(e, z). \tag{4.11}$$

Now, as $\mathcal{V}_{\tau h}^r$ is a subspace of $\mathcal{V}_\tau^r$, the first equation of (4.6) also holds for all functions in $\mathcal{V}_{\tau h}^r$. Therefore, substituting $v_{\tau h}$ in (4.6) and subtracting both equations, we obtain the so-called *Galerkin orthogonality*

$$B_{DG}^+(e, v_{\tau h}) = 0,\ \forall v_{\tau h} \in \mathcal{V}_{\tau h}^r. \tag{4.12}$$

Finally, subtracting (4.12) from (4.11), we obtain the *classical error representation*

$$L(e) = B^+_{DG}(e, z - v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}^r_{\tau h},$$

or equivalently employing the definition of problem (4.6)

$$L(e) = F(z - v_{\tau h}) - B^+_{DG}(u_{\tau h}, z - v_{\tau h}),\ \forall v_{\tau h} \in \mathcal{V}^r_{\tau h}.$$

□

Formulas (4.9) and (4.10) represent the error in the QoI as an integral over the whole space-time domain $\Omega \times I$. These quantities are usually bounded by the sum of local element contributions that will drive the goal-oriented adaptive process. However, they involve the analytical solutions of the primal and dual problems that are usually unknown. There exist several strategies in the literature to define computable local error contributions of equations (4.9) and (4.10).

In [131], authors employ the Backward Euler method in time and two levels of discretization is space for nonlinear time-dependent problems. Then, they develop a global-time space-adaptive algorithm by localizing the error estimator to the basis of each spatial mesh. Similarly in [149], the IMEX time-stepping scheme is considered with two levels of discretization both in space and time. Finally, the spatial and temporal error contributions are separated to perform GOA in both variables.

The strategy followed in [9] for hyperbolic problems consists in solving the primal and dual problems with a second order method in time such as Crank-Nicolson and then to perform local high-order post-processing both in space and time to approximate the dual solution. Then, they integrate by parts in space at every space-time cell and they select the free functions $v_{\tau h}$ in (4.10) as the interpolant of the dual solution in order to separate the space and time error components. A similar idea for parabolic problems is proposed in [54], where the authors perform a recovery strategy of the primal and dual solutions in the fine space-time mesh from the solution in the coarse mesh based on local computations. Here, the recovery is performed in patches of elements and independently in space and time.

Alternatively, authors in [28] propose an efficient blockwise goal-oriented adaptive algorithm for parabolic problems. Here, the idea is to employ non-uniform meshes that remain fixed for some periods of time (namely blocks) and solve the adjoint problem in the coarse scale. They present several strategies to select the length of the blocks to achieve a desired accuracy.

## 4.2. Error representation for the dPG formulation

We derive the error representation formulas (4.9) and (4.10) for the dPG formulation presented in Chapter 3. In the same way as in (4.5), we state that

$$B(u, v_\tau) + (u(0), v_\tau(0^-)) = B^-_{DG}(u, v_\tau),\ \forall v_\tau \in \mathcal{V}^r_\tau, \tag{4.13}$$

so we consider the following primal problems

$$\begin{cases} \text{Find } u \in \mathcal{U} \text{ and } u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \text{ such that} \\ B^-_{DG}(u, v_\tau) = F(v_\tau),\ \forall v_\tau \in \mathcal{V}^r_\tau, \\ B^-_{DG}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}. \end{cases} \tag{4.14}$$

Similarly to Section 4.1, we select the subspace $\mathcal{V}_\tau$ defined in (4.7) that is also the minimum subspace of $\mathcal{V}$ containing both $\mathcal{U}$ and $\mathcal{U}^q_\tau$. Therefore, taking into account that the jumps of $z$ at each time interface are zero, we consider

$$\begin{cases} \text{Find } z \in \mathcal{U} \text{ such that} \\ B^{*-}_{DG}(z, v_\tau) = L(v_\tau),\ \forall v_\tau \in \mathcal{V}_\tau, \end{cases} \tag{4.15}$$

where $B^{*-}_{DG}(\cdot,\cdot)$ is the adjoint operator of $B^-_{DG}(\cdot,\cdot)$ obtained after integration by parts in time.

Finally, we define the error of the primal problem as $e = u - u_{\tau h} \in \mathcal{V}_\tau$ and we represent the error in the QoI in the following theorem.

*Theorem* 4.2. (Error representation dPG) Let $B^-_{DG}(\cdot,\cdot)$ be the bilinear form defined in Eq. (3.5), $L(\cdot)$ the linear output functional defined in Eq. (4.1) and $e = u - u_{\tau h}$ the error of the primal problem. It holds that

$$L(e) = B^-_{DG}(e, z - v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}. \tag{4.16}$$

and equivalently

$$L(e) = F(z - v_{\tau h}) - B^-_{DG}(u_{\tau h}, z - v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}. \tag{4.17}$$

*Proof.* We have that $u \in \mathcal{U} \subset \mathcal{V}_\tau$ and $u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \subset \mathcal{U}^q_\tau \subset \mathcal{V}_\tau$ so $e = u - u_{\tau h} \in \mathcal{V}_\tau$. Substituting $e$ into (4.27) and integrating by parts in time, we obtain

$$L(e) = B^{*-}_{DG}(z, e) = B^-_{DG}(e, z). \tag{4.18}$$

Now, as $\tilde{\mathcal{V}}_{\tau h}$ is a subspace of $\mathcal{V}^r_\tau$, the first equation of (4.26) also holds for all functions in $\tilde{\mathcal{V}}_{\tau h}$. Therefore, substituting $v_{\tau h}$ in (4.26) and subtracting both equations, we obtain the Galerkin orthogonality

$$B^-_{DG}(e, v_{\tau h}) = 0,\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}. \tag{4.19}$$

Finally, subtracting (4.19) from (4.18) and employing the definition of problem (4.26), we obtain

$$L(e) = B^-_{DG}(e, z - v_{\tau h}) = F(z - v_{\tau h}) - B^-_{DG}(u_{\tau h}, z - v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}.$$

□

### 4.2.1. Effectivity index

We now calculate the effectivity index for two test cases

$$I_{eff} = \frac{Est}{L(e)}, \tag{4.20}$$

where $L(e)$ is the true error in the QoI and $Est$ is the estimated error when we approximate the exact primal and dual solutions in (4.16). It is desirable to have an effectivity index close to 1.

*Example* 4.1. In (2.1), we consider the linear diffusion equation (3.7) with $d = 1$, $\Omega = (0, 1)$, $T = 0.05$, $f = 0$, and $u_0(x) = \sin(\pi x)$. Then, the exact solution is

$$u(x, t) = e^{-\pi t} sin(\pi x).$$

We select the QoI defined in (4.1) with $g = 0$ and $z_T = \sin(\pi x)$, i.e.,

$$L(u) = (u(T), z_T).$$

In (4.16), we choose $z_{\tau h}$ as the solution of the dual problem in a coarse mesh and we approximate the exact primal and dual solutions in a finer mesh in space and time. That is, we consider

$$L(e) \sim Est = B^-_{DG}(e_{\tau h}, \varepsilon_{\tau h}),$$

where

$$e_{\tau h} = u_{\frac{\tau}{2}\frac{h}{2}} - u_{\tau h},\ \varepsilon_{\tau h} = z_{\frac{\tau}{2}\frac{h}{2}} - z_{\tau h}.$$

In Table 4.1 we show the effectivity index (4.20) when performing global space-time refinements. In order to fulfill the CFL condition, when we double the space elements in space we multiply by four the number of time steps. For the discretization in space, we employ a FEM with piecewise linear functions.

| Elements in space | Time steps | $I_{Eff}$ Forward Euler | $I_{Eff}$ Explicit Trapezoidal |
|---|---|---|---|
| 4 | 20 | 0.70978 | 0.75706 |
| 8 | 80 | 0.70366 | 0.75173 |
| 16 | 320 | 0.70212 | 0.75043 |
| 32 | 1,280 | 0.70174 | 0.75011 |
| 64 | 5,120 | 0.70164 | 0.75003 |
| 128 | 20,480 | 0.70162 | 0.75001 |

Table 4.1.: Effectivity index for the Forward Euler method and the explicit trapezoidal rule when performing uniform space-time refinements.

In time, we solve both primal and dual problems with the Forward Euler method and the explicit trapezoidal rule. Figure 4.1 shows the convergence of the true and the estimated error for both methods in time.

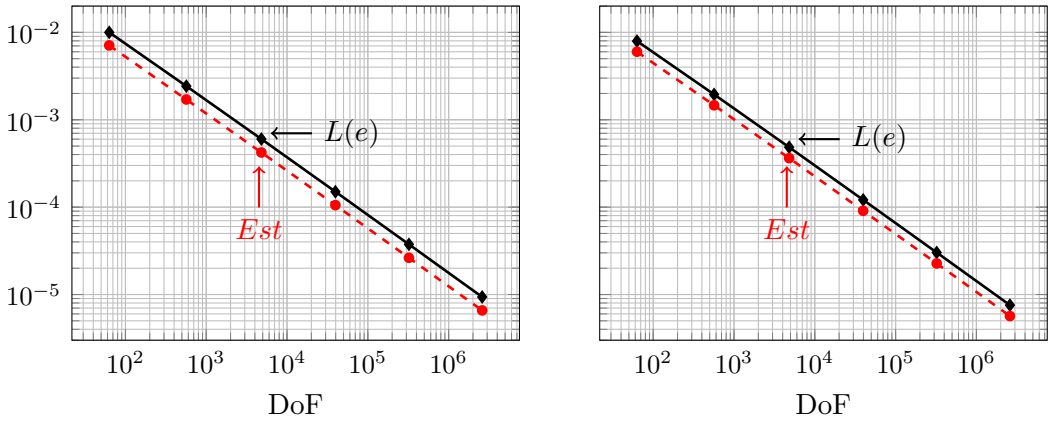



Figure 4.1.: Convergence of the true and the estimated errors for the Forward Euler method (left) and the explicit trapezoidal rule (right).

We conclude that, for this test case, the effectivity index for the Forward Euler method is $I_{eff} \sim 0.7$ and for the explicit trapezoidal rule is $I_{eff} \sim 0.75$.

*Example* 4.2. We now select an example with larger variations in time. We consider the linear diffusion equation (3.7) where $d = 1$, $\Omega = (0, 1)$, $T = 0.05$ and the initial condition $u_0$ and the source $f$ in such a way that the exact solution is

$$u(x, t) = \cos(100\pi t)\sin(\pi x).$$

In Table 4.2 we show the effectivity index (4.20) when performing global space-time refinements.

| Elements in space | Time steps | $I_{Eff}$ Forward Euler | $I_{Eff}$ Explicit Trapezoidal |
|---|---|---|---|
| 4 | 20 | 0.50483 | 0.73029 |
| 8 | 80 | 0.50073 | 0.73033 |
| 16 | 320 | 0.49943 | 0.73909 |
| 32 | 1,280 | 0.49908 | 0.74624 |
| 64 | 5,120 | 0.499 | 0.74896 |
| 128 | 20,480 | 0.49897 | 0.74973 |

Table 4.2.: Effectivity index for the Forward Euler method and the explicit trapezoidal rule when performing uniform space-time refinements.

Figure 4.2 shows the convergence of the true and the estimated error for the Forward Euler and the explicit trapezoidal methods.

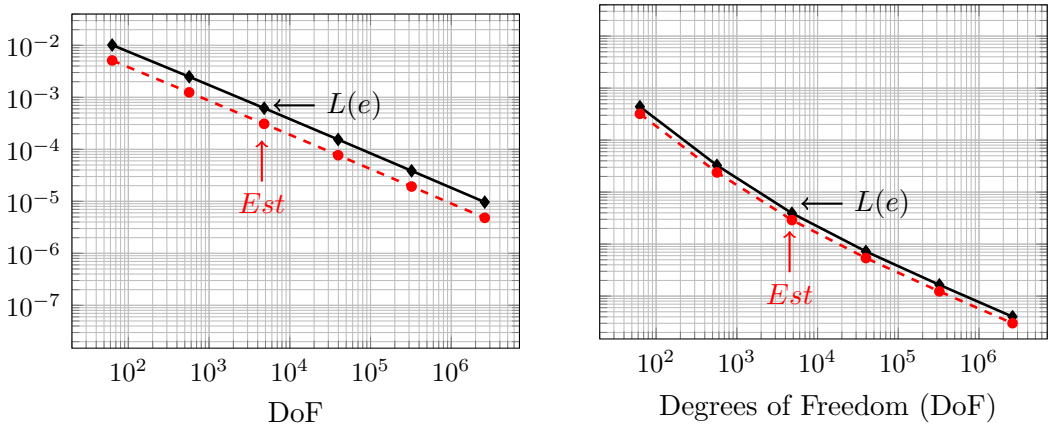



Figure 4.2.: Convergence of the true and the estimated errors for the Forward Euler method (left) and the explicit trapezoidal rule (right).

We conclude that, in this example, the effectivity index for the Forward Euler method is $I_{eff} \sim 0.5$ and for the explicit trapezoidal rule is $I_{eff} \sim 0.75$.

It seems the error representation (4.16) could be successfully employed to construct and apply explicit-in-time goal-oriented adaptive algorithms. Therefore, with the construction of the variational formulation presented in Chapter 3 and corresponding goal-oriented error representation described in this section, we could reproduce the majority of the existing goal-oriented adaptive algorithms using *explicit* RK methods in time to solve the primal and dual problems.

## 4.3. Discretization of the dual problem employing dPG formulation

The dPG formulation of problem (4.2) that we propose is

$$\begin{cases} \text{Find } z_\tau \in \tilde{\mathcal{V}}_\tau \text{ such that} \\ B^{*-}_{DG}(z_\tau, v_\tau) = L(v_\tau),\ \forall v_\tau \in \tilde{\mathcal{U}}_\tau, \end{cases} \tag{4.21}$$

where $\tilde{\mathcal{U}}_\tau$ and $\tilde{\mathcal{V}}_\tau$ are the spaces defined in (3.3). Here, $B^{*-}_{DG}(\cdot,\cdot)$ is the resulting bilinear form after integrating by parts the time derivative and the space advection terms of the form $B^{-}_{DG}(\cdot,\cdot)$ (see Appendix B.2 for details)

$$\begin{aligned} B^{*-}_{DG}(z,v) := & \sum_{k=1}^{m} \int_{I_k} \Big( -\langle z_t, v\rangle + (\nu\nabla z, \nabla v) - (\beta\cdot\nabla z, v)\Big) dt \\ & - \sum_{k=1}^{m} \big( [\![z]\!]^{k-1}, v(t^+_{k-1})\big) + (z(T^-), v(T^+)). \end{aligned}$$

Now, we select the same subspaces as in the primal problem defined in (3.6) and we introduce the fully discrete problem

$$\begin{cases} \text{Find } z_{\tau h} \in \tilde{\mathcal{V}}_{\tau h} \text{ such that} \\ B^{*-}_{DG}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}),\ \forall v_{\tau h} \in \tilde{\mathcal{U}}_{\tau h}. \end{cases} \tag{4.22}$$

In (4.22), we shift the trial and test spaces for the fully discrete dual problem with respect to those of the fully discrete primal problem (3.6).

### 4.3.1. Forward Euler method backwards in time

We select piecewise constant functions ($q = r = 0$) in (4.22). Therefore, the spaces in (4.22) are the ones we defined in (3.8) and we express the solution $z_{\tau h}$ as

$$z_{\tau h}(\mathbf{x}, t) = \sum_{k=0}^{m} z_h^k(\mathbf{x})\varphi^k(t), \tag{4.23}$$

where $z_h^k \in V_h^k,\ \forall k = 0, \dots, m$ and $\varphi^k(t)$ are the functions defined in (2.13).

We select the following test functions (see Figure 4.3)

$$v_h^{k-1}(\mathbf{x})\phi^{k-1}(t),\ \forall k = 1, \dots, m+1, \tag{4.24}$$

where $v_h^{k-1} \in V_h^{k-1},\ \forall k = 1, \dots, m+1$ and $\phi^{k-1}(t)$ are the functions defined in (3.10).

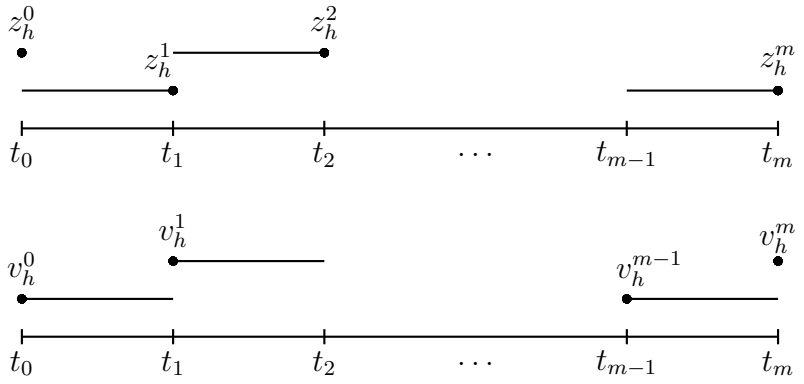



Figure 4.3.: Trial and test functions for the dual problem when $q = r = 0$.

*Theorem* 4.3. (Backwards-in-time Forward Euler method) Selecting trial and test functions as in (4.23) and (4.24), problem (4.22) leads to the following scheme

$$\begin{cases} \text{Find } z_h^m \in V_h^m \text{ and } z_h^{k-1} \in V_h^{k-1},\ \forall k = m, \dots, 1, \text{ such that} \\ \\ \left(z_h^{k-1}, v_h^{k-1}\right) = \left(z_h^k, v_h^{k-1}\right) - \tau_k \left(\nu \nabla z_h^k, \nabla v_h^{k-1}\right) + \tau_k \left(\beta \cdot \nabla z_h^k, v_h^{k-1}\right) \\ \qquad\qquad + \displaystyle\int_{I_k} \left\langle g, v_h^{k-1} \right\rangle dt,\ \forall v_h^{k-1} \in V_h^{k-1}, \\ \\ \left(z_h^m, v_h^m\right) = \left(z_T, v_h^m\right),\ \forall v_h^m \in V_h^m, \end{cases} \tag{4.25}$$

which is also a variant of the Forward Euler method but running backwards in time.

*Proof.* From (4.24), the last test function is $v_{\tau h}(t) = v_h^m \phi^m(t)$. Substituting it and the solution (4.23) on (4.22), we obtain

$$\left(z_h^m, v_h^m\right) = \left(z_T, v_h^m\right),\ \forall v_h^m \in V_h^m.$$

Then, each test function $v_{\tau h}(t) = v_h^{k-1}\phi^{k-1}(t),\ \forall k = 1, \dots, m$, has local support in $[t_{k-1}, t_k)$, and problem (4.22) reads

$$\begin{aligned} -\int_{I_k} \left\langle z_{\tau h,t}, v_{\tau h} \right\rangle dt + \int_{I_k} (\nu \nabla z_{\tau h}, \nabla v_{\tau h}) dt - \int_{I_k} (\beta \cdot \nabla z_{\tau h}, v_{\tau h}) dt \\ - \left(\left[\!\left[ z_{\tau h} \right]\!\right]^{k-1}, v_{\tau h}(t_{k-1}^+)\right) = \int_{I_k} \left\langle g, v_{\tau h} \right\rangle dt. \end{aligned}$$

Now, each term of the previous formula becomes

$$\int_{I_k} \left\langle z_{\tau h,t}, v_{\tau h} \right\rangle dt = \left(z_h^k, v_h^{k-1}\right) \int_{I_k} \left(\frac{\partial}{\partial t}\varphi^k\right)\phi^{k-1} dt = 0,$$

$$\int_{I_k}(\nu\nabla z_{\tau h},\nabla v_{\tau h})dt=(\nu\nabla z_h^k,\nabla v_h^{k-1})\int_{I_k}\varphi^k\phi^{k-1}dt=\tau_k(\nu\nabla z_h^k,\nabla v_h^{k-1}),$$

$$\int_{I_k}(\beta\cdot\nabla z_{\tau h},v_{\tau h})dt=(\beta\cdot\nabla z_h^k,v_h^{k-1})\int_{I_k}\varphi^k\phi^{k-1}dt=\tau_k(\beta\cdot\nabla z_h^k,v_h^{k-1}),$$

$$([\![z_{\tau h}]\!]^{k-1},v_{\tau h}(t_{k-1}^+))=(z_h^k\varphi^k(t_{k-1}^+)-z_h^{k-1}\varphi^{k-1}(t_{k-1}^-),v_h^{k-1}\phi^{k-1}(t_{k-1}^+))=(z_h^k-z_h^{k-1},v_h^{k-1}),$$

$$\int_{I_k}\langle g,v_{\tau h}\rangle\,dt=\int_{I_k}\left\langle g,v_h^{k-1}\right\rangle dt.$$

Finally, we obtain

$$\left(z_h^{k-1},v_h^{k-1}\right)=\left(z_h^k,v_h^{k-1}\right)-\tau_k\left(\nu\nabla z_h^k,\nabla v_h^{k-1}\right)+\tau_k\left(\beta\cdot\nabla z_h^k,v_h^{k-1}\right)+\int_{I_k}\left\langle g,v_h^{k-1}\right\rangle dt.$$

□

As for the primal problem, to obtain the classical forward Euler method, we can express $g(\mathbf{x},t)$ as

$$g(\mathbf{x},t)=\sum_{k=0}^{m}g^k(\mathbf{x})\varphi^k(t),$$

where $g^k(\mathbf{x}):=g(\mathbf{x},t_k)$ and the source term in (4.22) becomes

$$\int_{I_k}\left\langle g,v_h^{k-1}\right\rangle dt=\int_{I_k}\left\langle g^k,v_h^{k-1}\right\rangle dt=\tau_k\left\langle g^k,v_h^{k-1}\right\rangle.$$

## 4.4. Discrete error representation

Following a similar strategy as in [131], we approximate the exact solutions $u$ and $z$ of the primal and dual problems by enriching the subspaces $\tilde{\mathcal{U}}_{\tau h}$ and $\tilde{\mathcal{V}}_{\tau h}$ defined in (3.8) with $q=r=0$ and selecting

$$u\simeq u_{\tau\frac{h}{2}}\in\tilde{\mathcal{U}}_{\tau\frac{h}{2}},\quad z\simeq z_{\tau\frac{h}{2}}\in\tilde{\mathcal{V}}_{\tau\frac{h}{2}},$$

where $\tilde{\mathcal{U}}_{\tau\frac{h}{2}}$ and $\tilde{\mathcal{V}}_{\tau\frac{h}{2}}$ are the subspaces obtained from splitting in half each spatial element of $\tilde{\mathcal{U}}_{\tau h}$, $\tilde{\mathcal{V}}_{\tau h}$, respectively. We do not consider a finer space in time since in our adaptive strategy we are only interested on representing the error induced by a poor space discretization.

We define the following discrete errors

$$e_{\tau h}:=u_{\tau\frac{h}{2}}-u_{\tau h},\quad \varepsilon_{\tau h}:=z_{\tau\frac{h}{2}}-z_{\tau h}.$$

and we approximate the exact error as

$$e = u - u_{\tau h} \sim u_{\tau \frac{h}{2}} - u_{\tau h} = e_{\tau h},$$

$$\varepsilon = z - z_{\tau h} \sim z_{\tau \frac{h}{2}} - z_{\tau h} = \varepsilon_{\tau h}.$$

We define the approximated errors at each time step as

$$e_h^k = u_{\frac{h}{2}}^k - u_h^k, \;\; \varepsilon_h^k = z_{\frac{h}{2}}^k - z_h^k, \; \forall k = 0, \dots, m.$$

We now focus on reducing the error in the QoI coming from the spatial discretization $L\left(e_{\tau h}\right)$. Hence, we derive a discrete version of Theorem 4.2 with the fully discrete spaces we defined in (3.8). To do so, we solve the primal and dual problems at two levels of discretization, i.e.,

$$\begin{cases} \text{Find } u_{\tau \frac{h}{2}} \in \tilde{\mathcal{U}}_{\tau \frac{h}{2}} \text{ and } u_{\tau h} \in \tilde{\mathcal{U}}_{\tau h} \text{ such that} \\ B_{DG}^{-}(u_{\tau \frac{h}{2}}, v_{\tau \frac{h}{2}}) = F(v_{\tau \frac{h}{2}}), \; \forall v_{\tau \frac{h}{2}} \in \tilde{\mathcal{V}}_{\tau \frac{h}{2}}, & \text{(4.26a)} \\ B_{DG}^{-}(u_{\tau h}, v_{\tau h}) = F(v_{\tau h}), \; \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}, & \text{(4.26b)} \end{cases}$$

$$\begin{cases} \text{Find } z_{\tau \frac{h}{2}} \in \tilde{\mathcal{V}}_{\tau \frac{h}{2}} \text{ and } z_{\tau h} \in \tilde{\mathcal{V}}_{\tau h} \text{ such that} \\ B_{DG}^{*-}(z_{\tau \frac{h}{2}}, v_{\tau \frac{h}{2}}) = L(v_{\tau \frac{h}{2}}), \; \forall v_{\tau \frac{h}{2}} \in \tilde{\mathcal{U}}_{\tau \frac{h}{2}}, & \text{(4.27a)} \\ B_{DG}^{*-}(z_{\tau h}, v_{\tau h}) = L(v_{\tau h}), \; \forall v_{\tau h} \in \tilde{\mathcal{U}}_{\tau h}, & \text{(4.27b)} \end{cases}$$

As the discrete solutions are piecewise constant in time, and taking into account the discretization we select in (3.9) and (4.23), we simplify the bilinear forms $B_{DG}^{-}(\cdot,\cdot)$ and $B_{DG}^{*-}(\cdot,\cdot)$ at the discrete level as

$$\begin{aligned} B_{DG}^{-}(u_{\tau h}, v_{\tau h}) &= \sum_{k=1}^{m} \left(u_h^k - u_h^{k-1}, v_h^k\right) + \tau_k \left(\nu \nabla u_h^{k-1}, \nabla v_h^k\right) \\ &\quad + \tau_k \left(\beta \cdot \nabla u_h^{k-1}, v_h^k\right) + \left(u_h^0, v_h^0\right), \\ B_{DG}^{*-}(z_{\tau h}, v_{\tau h}) &= \sum_{k=1}^{m} \left(z_h^{k-1} - z_h^k, v_h^{k-1}\right) + \tau_k \left(\nu \nabla z_h^k, \nabla v_h^{k-1}\right) \\ &\quad - \tau_k \left(\beta \cdot \nabla z_h^k, v_h^{k-1}\right) + \left(z_h^m, v_h^m\right). \end{aligned} \tag{4.28}$$

*Theorem* 4.4. (Discrete error representation dPG). Let $B_{DG}^{-}(\cdot,\cdot)$ be the bilinear form defined in Eq. (3.5), $L(\cdot)$ the output functional defined in Eq. (4.1), and $e_{\tau h} = u_{\tau \frac{h}{2}} - u_{\tau h}$ and $\varepsilon_{\tau h} = z_{\tau \frac{h}{2}} - z_{\tau h}$ the discrete errors of the primal and dual problems, respectively. It holds that

$$L\left(e_{\tau h}\right) = B_{DG}^{-}\left(e_{\tau h}, \varepsilon_{\tau h}\right). \tag{4.29}$$

*Proof.* Since $e_{\tau h} = u_{\tau \frac{h}{2}} - u_{\tau h} \in \tilde{\mathcal{U}}_{\tau \frac{h}{2}}$, we substitute $e_{\tau h}$ into (4.27a) and we have that

$$L\left(e_{\tau h}\right) = B^{*-}_{DG}(z_{\tau \frac{h}{2}}, e_{\tau h}),$$

and equivalently

$$L\left(e_{\tau h}\right) = \sum_{k=1}^{m} \left(z^{k-1}_{\frac{h}{2}} - z^{k}_{\frac{h}{2}}, e^{k-1}_{h}\right) + \tau_k \left(\nu \nabla z^{k}_{\frac{h}{2}}, \nabla e^{k-1}_{h}\right) - \tau_k \left(\beta \cdot \nabla z^{k}_{\frac{h}{2}}, e^{k-1}_{h}\right) + \left(z^{m}_{\frac{h}{2}}, e^{m}_{h}\right). \tag{4.30}$$

Now, we reorganize the first and the last terms of (4.30)

$$\begin{aligned}
&\sum_{k=1}^{m} \left(z^{k-1}_{\frac{h}{2}} - z^{k}_{\frac{h}{2}}, e^{k-1}_{h}\right) + \left(z^{m}_{\frac{h}{2}}, e^{m}_{h}\right) \\
&= \sum_{k=0}^{m-1} \left(z^{k}_{\frac{h}{2}}, e^{k}_{h}\right) - \sum_{k=1}^{m} \left(z^{k}_{\frac{h}{2}}, e^{k-1}_{h}\right) + \left(z^{m}_{\frac{h}{2}}, e^{m}_{h}\right) \\
&= \sum_{k=1}^{m} \left(z^{k}_{\frac{h}{2}}, e^{k}_{h}\right) - \sum_{k=1}^{m} \left(z^{k}_{\frac{h}{2}}, e^{k-1}_{h}\right) + \left(z^{0}_{\frac{h}{2}}, e^{0}_{h}\right) \\
&= \sum_{k=1}^{m} \left(z^{k}_{\frac{h}{2}}, e^{k}_{h} - e^{k-1}_{h}\right) + \left(z^{0}_{\frac{h}{2}}, e^{0}_{h}\right),
\end{aligned}$$

and by integrating the advection term in (4.30) by parts in space, we get

$$L\left(e_{\tau h}\right) = \sum_{k=1}^{m} \left(z^{k}_{\frac{h}{2}}, e^{k}_{h} - e^{k-1}_{h}\right) + \tau_k \left(\nu \nabla z^{k}_{\frac{h}{2}}, \nabla e^{k-1}_{h}\right) + \tau_k \left(z^{k}_{\frac{h}{2}}, \beta \cdot \nabla e^{k-1}_{h}\right) + (z^{0}_{\frac{h}{2}}, e^{0}_{h}),$$

then, we have

$$L\left(e_{\tau h}\right) = B^{-}_{DG}(e_{\tau h}, z_{\tau \frac{h}{2}}). \tag{4.31}$$

Since $\tilde{\mathcal{V}}_{\tau h}$ is a subspace of $\tilde{\mathcal{V}}_{\tau \frac{h}{2}}$ and following the same argument as for the continuous level, we express the *discrete Galerkin orthogonality* as

$$B^{-}_{DG}(e_h, v_{\tau h}) = 0, \ \forall v_{\tau h} \in \tilde{\mathcal{V}}_{\tau h}, \tag{4.32}$$

and finally, subtracting (4.32) from (4.31) with $z_{\tau h} \in \mathcal{V}_{\tau h}$ we obtain

$$L\left(e_{\tau h}\right) = B^{-}_{DG}(e_{\tau h}, z_{\tau \frac{h}{2}}) - B^{-}_{DG}\left(e_{\tau h}, z_{\tau h}\right) = B^{-}_{DG}(e_{\tau h}, \varepsilon_{\tau h}).$$

□

*Remark* 4.1. In problems (3.1) and (4.3), if we employ the Spectral Element Method (SEM) in space, we obtain diagonal mass matrices for each time step, which is computationally cheaper to solve than employing the finite element method. However, the spectral element method does not integrate exactly the $L^2$-terms of the operator $B^{-}_{DG}(\cdot, \cdot)$, and therefore, we loose the equality of Theorem 4.29. Nonetheless, we could still use it to approximate the error representation.

*Remark* 4.2. The development of a discrete error representation similar to (4.29) for the higher-order explicit RK methods defined in Chapter 3 could be challenging. The reason is that we need to ensure some belonging relationships between the trial and test spaces and we are dealing with incomplete spaces. In this dissertation, we focus in the lowest order case in time (Forward Euler method) and we will consider the higher-order case as a future work.

## 4.5. GOA in space for explicit-in-time methods

### 4.5.1. GOA in space

The goal-oriented adaptive strategy in space we propose is based on the error representation (4.29)

$$\begin{aligned} L\left(e_{\tau h}\right) = &\left(e_h^0, \varepsilon_h^0\right) + \sum_{k=1}^{m} \left(e_h^k - e_h^{k-1}, \varepsilon_h^k\right) \\ &+ \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k\right) + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k\right), \end{aligned}$$

or equivalently

$$\begin{aligned} L\left(e_{\tau h}\right) = &\sum_{i=1}^{n_0} \left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0} + \sum_{k=1}^{m} \sum_{i=1}^{n_k} \left(e_h^k - e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k} \\ &+ \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k}, \end{aligned} \tag{4.33}$$

where $\{\Omega_i^k\}_{i=1,\dots,n_k},\ \forall k = 0, \dots, m$ is a partition of the spatial domain $\Omega$ at $t = t_k$ and $(\cdot,\cdot)_{\Omega_i^k}$ is the restriction of the inner product in $L^2(\Omega)$ to each element $\Omega_i^k$.

Finally, applying the triangle inequality in (4.33), we obtain the following upper bound of the error in the QoI

$$\begin{aligned} \left|L\left(e_{\tau h}\right)\right| \leqslant &\sum_{i=1}^{n_0} \left|\left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0}\right| + \sum_{k=1}^{m} \sum_{i=1}^{n_k} \Big|\left(e_h^k - e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k} \\ &+ \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k}\Big|, \end{aligned} \tag{4.34}$$

which we use to guide the goal-oriented adaptive process.

We define the error estimator of each time step as

$$Est_k := \sum_{i=0}^{n_k} \eta_i^k,\ \forall k = 0, \dots, m, \tag{4.35}$$

where

$$\begin{aligned}\eta_i^0 &:= \left(e_h^0, \varepsilon_h^0\right)_{\Omega_i^0}, \\ \eta_i^k &:= \left(e_h^k - e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\nu \nabla e_h^{k-1}, \nabla \varepsilon_h^k\right)_{\Omega_i^k} + \tau_k \left(\beta \cdot \nabla e_h^{k-1}, \varepsilon_h^k\right)_{\Omega_i^k},\end{aligned} \tag{4.36}$$

are the error estimators of each spatial element $\Omega_i^k$ at each time step.

We employ error estimators (4.35) and (4.36) to decide which space-time elements we need to refine in space in order to reduce the error in the QoI.

### 4.5.2. Time adaptivity based on the CFL condition

As we explained in Chapter 3, scheme (3.1) is explicit so it is conditionally stable. Similarly, scheme (4.3) is also explicit and, therefore, the CFL condition must be satisfied in both problems to ensure the stability of the discretization. Even if we solve both problems in the same space-time grid, the information is propagated in opposite directions, as the dual problem runs backwards in time. Figure 4.4 illustrates the influence area of a space-time point and the direction of the information of each problem. Schemes (3.1) and (4.3) must capture those areas of influence. Hence, we have to address two CFL conditions: one for the primal problem and another for the dual problem.

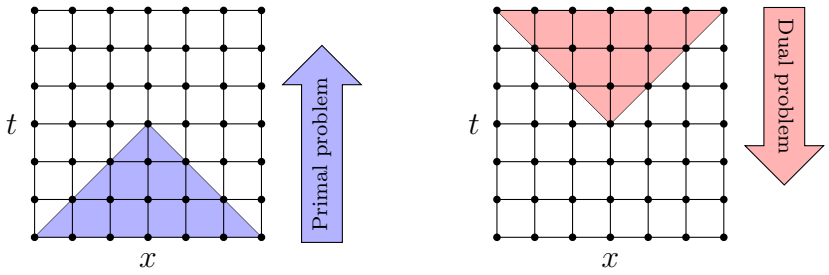



Figure 4.4.: Transmission of the information for the primal and dual problems.

If we denote by $h_k$ the minimum element size of the spatial mesh at $t_k$, the two CFL conditions are

$$\frac{\tau_k}{h_k^2}\nu < \frac{1}{6} \text{ (dual)}, \quad \frac{\tau_{k+1}}{h_k^2}\nu < \frac{1}{6} \text{ (primal)}, \tag{4.37}$$

where $\tau_k = t_k - t_{k-1}$ and $\tau_{k+1} = t_{k+1} - t_k$.

Therefore, we adapt the time grid based on the CFL conditions defined in (4.37). Once the spatial meshes are refined, we identify those time intervals where the CFL conditions are not satisfied and we split them by introducing new synchronous levels of spatial discretization (spatial meshes). Figure 4.5 shows the

adaptive process in time. We have $h_k^2$ in (4.37), thus we need to split the time step by four in order to satisfy the conditions.

There are different options to select the spatial meshes we insert in a time interval. In Figure 4.5 we employ the same spatial mesh as the one associated to $t_{k-1}$, but it could be the one from $t_k$ or, for example, the union of the spaces associated to both meshes. We obtain similar results in Section 4.6 by employing different approaches.

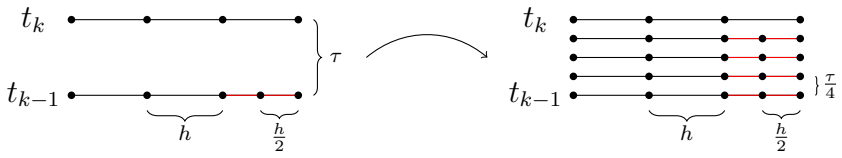



Figure 4.5.: Time refinements based on the CFL condition.

### 4.5.3. Algorithm

Algorithm 1 describes our goal-oriented adaptive process in space together with a CFL-based adaptive process in time.

The input arguments of Algorithm 1 are the time grid $\{\tau_k\}_{k=1,\dots,m}$, the spatial mesh at each time step $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, the tolerance $tol$ for the error in the QoI and two parameters $\theta, \lambda \in [0,1]$.

We first calculate the primal solutions $u_{\tau h}$ and $u_{\tau \frac{h}{2}}$ forwards in time and we compute the dual solutions $z_{\tau h}$ and $z_{\tau \frac{h}{2}}$ backwards in time. Then, we estimate $Est_k$ and $\eta_i^k$ and for all spatial meshes satisfying $|Est_k| \geqslant \theta \cdot \max\limits_{0 \leqslant k \leqslant m} |Est_k|$, we refine those elements in space that satisfy $|\eta_i^k| \geqslant \lambda \cdot \max\limits_{1 \leqslant i \leqslant n_k} |\eta_i^k|$. Finally, we refine those time intervals where the CFL conditions (4.37) are not satisfied. The process ends when the relative error

$$QoI := \frac{|L(e_{\tau h})|}{\left|L\left(u_{\tau \frac{h}{2}}\right)\right|} \cdot 100,$$

is below a user-prescribed fixed tolerance $tol$.

*Remark* 4.3. In Algorithm 1, we have two meshes at each time step: the fine mesh and the coarse mesh, $\mathcal{M}_{\frac{h}{2}}^k$ and $\mathcal{M}_h^k$, respectively. The errors $e_h^k$ and $\varepsilon_h^k$ are computed in the fine mesh $\mathcal{M}_{\frac{h}{2}}^k$, whereas the estimators $\eta_i^k$ are calculated over each element of the coarse mesh $\mathcal{M}_h^k$.

**Algorithm 1** Goal-oriented in space and CFL-based in time adaptivity

1: **Input:** $\{\tau_k\}_{k=1,\dots,m}$, $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, $tol$, $\theta$, $\lambda$
2: Compute $u_{\tau h}$, $u_{\tau\frac{h}{2}}$ and $e_{\tau h}$
3: Compute $z_{\tau h}$, $z_{\tau\frac{h}{2}}$ and $\varepsilon_{\tau h}$
4: Estimate $QoI$
5: **while** $QoI > tol$ **do**
6:  **for** $k = 0$ to $m$ and $i = 1$ to $n$ **do**
7:  Compute $\eta_i^k$ and $Est_k$
8:  **end for**
9:  **for** $k = 0$ to $m$ and $i = 1$ to $n$ **do**
10:  **if** $|Est_k| \geqslant \theta \cdot \max\limits_{0\leqslant k\leqslant m} |Est_k|$ and $|\eta_i^k| \geqslant \lambda \cdot \max\limits_{1\leqslant i\leqslant n_k} |\eta_i^k|$ **then**
11:  Refine element $\Omega_i^k$ of $\mathcal{M}_h^k$
12:  **end if**
13:  **end for**
14:  **for** $k = 1$ to $m$ **do**
15:  **if** CFL-conditions are not satisfied in $I_k$ **then**
16:  Refine interval $I_k$
17:  **end if**
18:  **end for**
19:  Compute $u_{\tau h}$, $u_{\tau\frac{h}{2}}$ and $e_{\tau h}$
20:  Compute $z_{\tau h}$, $z_{\tau\frac{h}{2}}$ and $\varepsilon_{\tau h}$
21:  Estimate $QoI$
22: **end while**

*Remark* 4.4. In Algorithm 1 (line 10), we employ the maximum strategy [143]. Graphically, we can interpret this method as in Figure 4.6. First, we order, for example, all the estimators in time from highest to lowest. Then, we draw a horizontal line at $\theta \cdot \max\limits_{0\leqslant k\leqslant m} |Est_k|$. Finally, we refine those spatial meshes whose estimators are above the line.

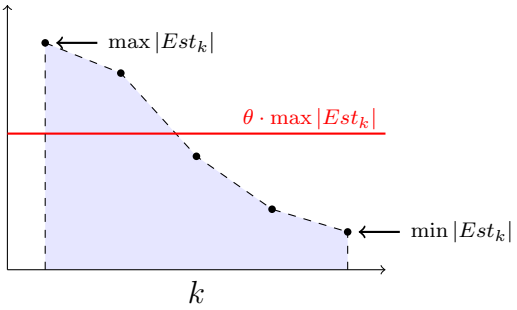



Figure 4.6.: Maximum strategy.

## 4.6. Numerical results

### 4.6.1. Diffusion problem

In (2.1), we consider $d = 1$, $\Omega = (0, 1)$, $T = 2$, $\beta = 0$, $f(x, t) = (1 + \pi^2 t)\sin(\pi x)$ and $u(x, 0) = 0$. We select a discontinuous diffusion coefficient

$$\nu(x) = \begin{cases} 0.01, & x \in [0.25, 0.75], \\ 0.001, & \text{elsewhere}, \end{cases}$$

and the following output functional

$$L(u) = \int_I \int_{\Omega_0} u(x, t) dx dt,$$

where $\Omega_0 = (0, 0.25) \cup (0.75, 1)$.

Figure 4.7 shows the primal (3.6) and dual (4.22) reference solutions. In Figure 4.8 (left) we can see the relative error in the QoI and the upper bound (4.34) when we perform four uniform space-time refinements. We start with $2^3$ elements in space and 40 time steps for the first iteration. Then, we split each time interval four times by applying uniform refinements in space in order to satisfy the CFL condition. We calculate the total number of DoF as the sum of the degrees of freedom of each spatial mesh, i.e.,

$$DoF = \sum_{k=0}^{m} DoF(\Omega^k).$$

In Algorithm 1 we set 40 time steps, $2^3$ elements in space, $\theta = \lambda = 0.3$ and $tol = 0.01\%$. In the first iteration, we select the number of time steps in such a way that we obtain limit values of the CFL conditions. Then, the algorithm is going to refine those time intervals where the CFL violated.

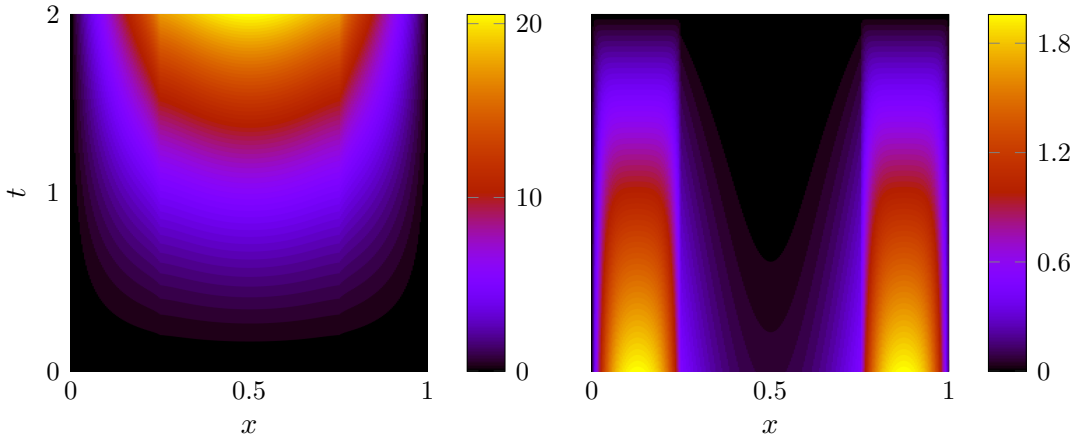



Figure 4.7.: Solution of the primal (left) and dual (right) problems.

Figure 4.9 exhibits the adapted space-time mesh after 28 iterations and Figure 4.8 (right) shows the relative error in the QoI and the upper bound (4.34). As a comparison, for uniform refinements we need more than $10^5$ degrees of freedom to achieve a relative error of 0.01%, while for the adaptive method we need around $5 \cdot 10^3$.

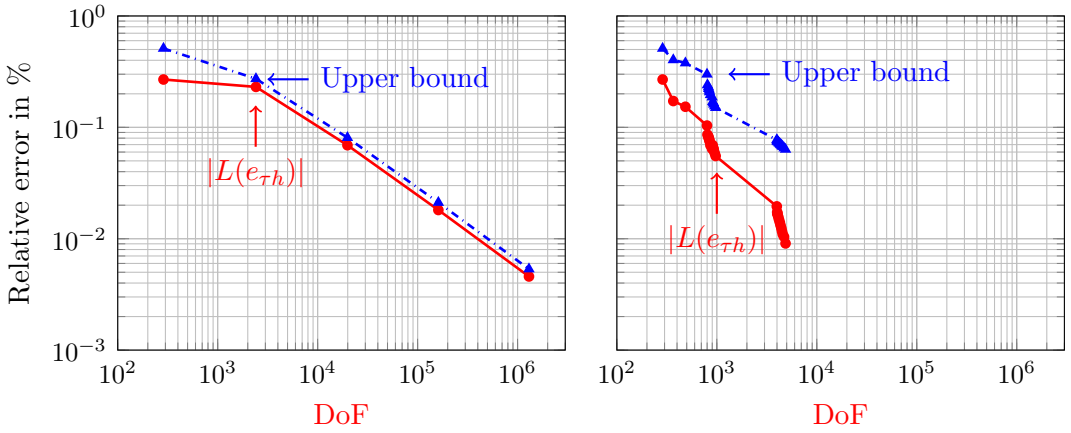



Figure 4.8.: Error in the QoI and upper bound (4.34) for uniform space-time refinements (left) and Algorithm 1 (right).

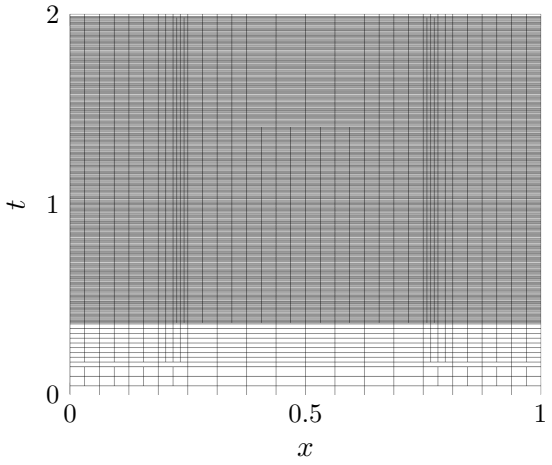



Figure 4.9.: Adapted space-time mesh.

### 4.6.2. Advection-diffusion problem

Let $d = 1$, $\Omega = (0, 1)$ and $T = 0.5$. We consider problem (2.1) with $\nu = 0.015$, $\beta = 1.5$, $f(x, t) = 0$,

$$u_0(x) = \begin{cases} 1, & x \in [0.125, 0.375], \\ 0, & \text{elsewhere}, \end{cases}$$

and we select the following output functional

$$L(u) = \int_{I_0} \int_{\Omega_0} u(x, t) dx dt,$$

where $\Omega_0 \times I_0 = (0.75, 1) \times (0.4, 0.5]$ is a subdomain of $\Omega \times I$.

Figure 4.10 shows the reference solutions of the primal (3.6) and dual (4.22) problems. We observe that in the primal problem, the initial condition is propagated. Due to the boundary conditions, a boundary layer is formed in the final time steps at the right endpoint of the spatial domain. We are interested in reducing the error of the solution in such boundary layer.

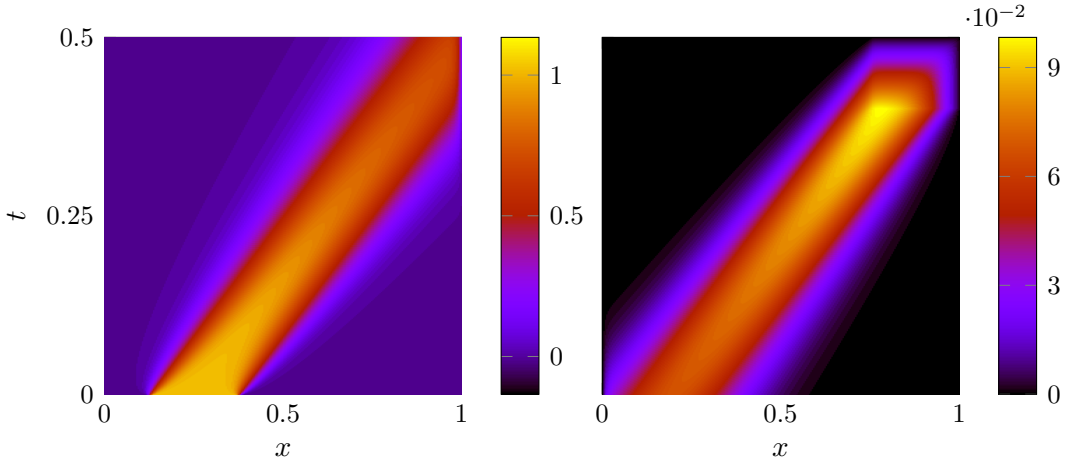



Figure 4.10.: Solution of the primal (left) and dual (right) problems.

Figure 4.11 (left) shows the relative error in the QoI and the upper bound (4.34) when we perform four uniform space-time refinements starting with 15 time steps and $2^3$ elements in space.

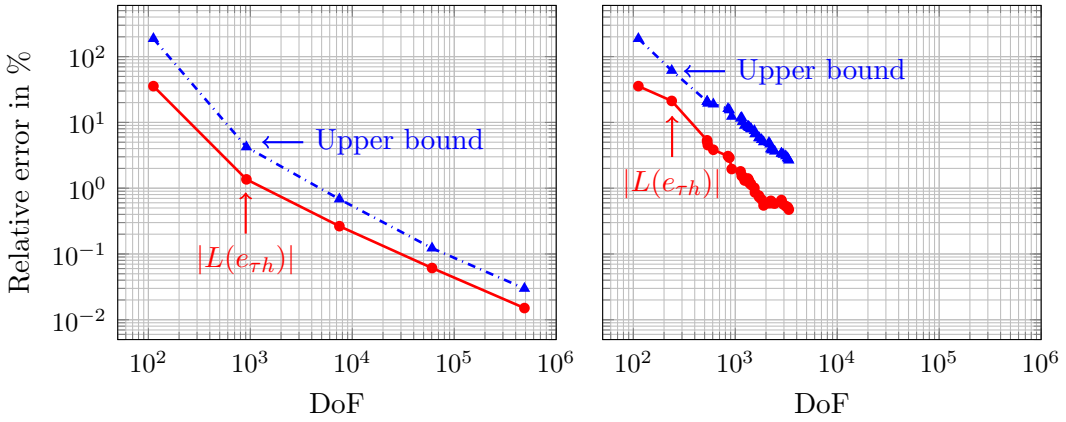



Figure 4.11.: Error in the QoI and upper bound (4.34) for uniform space-time refinements (left) and Algorithm 1 (right).

We again set $\theta = \lambda = 0.3$ and $tol = 0.5\%$ in Algorithm 1. Figure 4.11 (right) exhibits the relative error in the QoI and the upper bound when we perform adaptivity. Figure 4.12 shows the adapted space-time mesh after 37 iterations. We conclude that the convergence ratio in this case is similar to uniform space-time refinements as we are not heavily employing local space-time refinements.

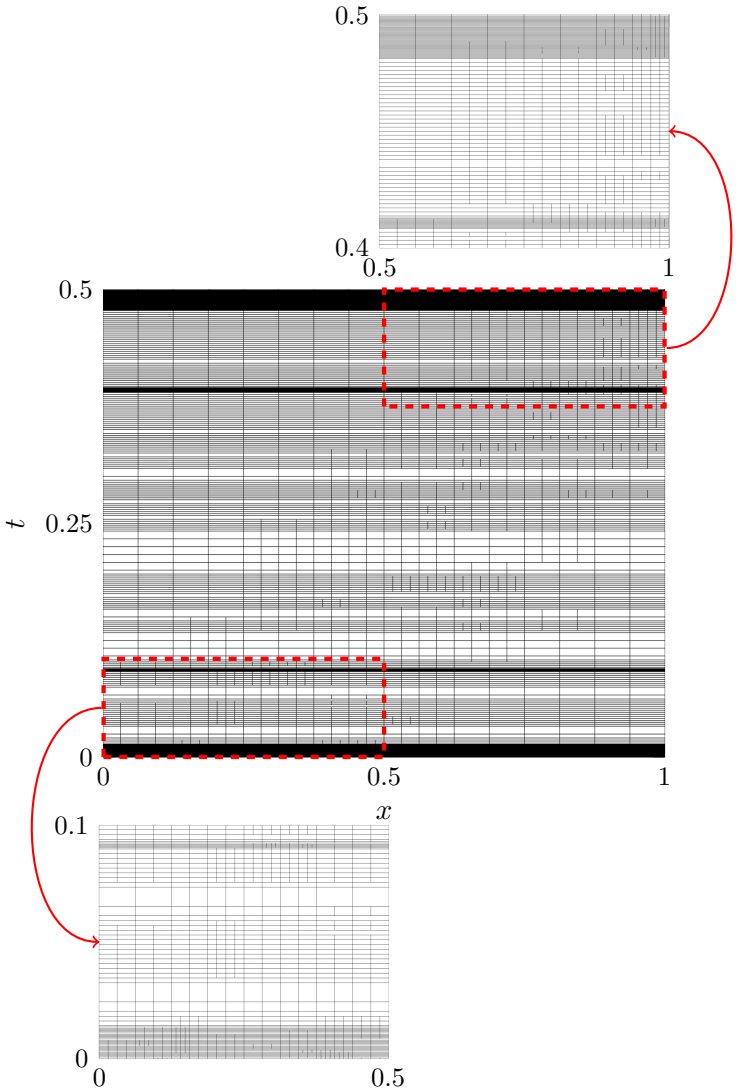



Figure 4.12.: Adapted space-time mesh.

### 4.6.3. Symmetric estimator

We consider $d = 1$, $\Omega = (0, 1)$, $T = 0.5$, $\nu = 0.015$, $\beta = 0$, $f(x, t) = 0$ and

$$u_0(x) = \begin{cases} 1, & x \in [0.125, 0.375], \\ 0, & \text{elsewhere}, \end{cases}$$

and we select the following output functional

$$L(u) = \int_{\Omega_0} u(x, T)dx,$$

where $\Omega_0 = (0.625, 0.875) \subset \Omega$. Figure 4.13 shows the primal and dual reference solutions where both solutions are symmetric and they run in opposite directions in time. Figure 4.14 exhibits the relative error in the QoI and the upper bound after 4 uniform space-time refinements starting from $2^3$ elements in space and 15 time steps.

For Algorithm 1 we again set $\theta = \lambda = 0.3$ and $tol = 3\%$. Employing the operator $B^-_{DG}(\cdot, \cdot)$ as the adaptive criteria, the algorithm tends to refine the mesh to reduce de error of the primal problem. Analogously, if we employ $B^{*-}_{DG}(\cdot, \cdot)$, then the algorithm applies refinements to reduce the error of the dual problem. Therefore, to obtain a symmetric adapted mesh we employ the following symmetric error representation

$$L(e_{\tau h}) = \frac{1}{2}B^-_{DG}(e_{\tau h}, \varepsilon_{\tau h}) + \frac{1}{2}B^{*-}_{DG}(\varepsilon_{\tau h}, e_{\tau h}). \tag{4.38}$$

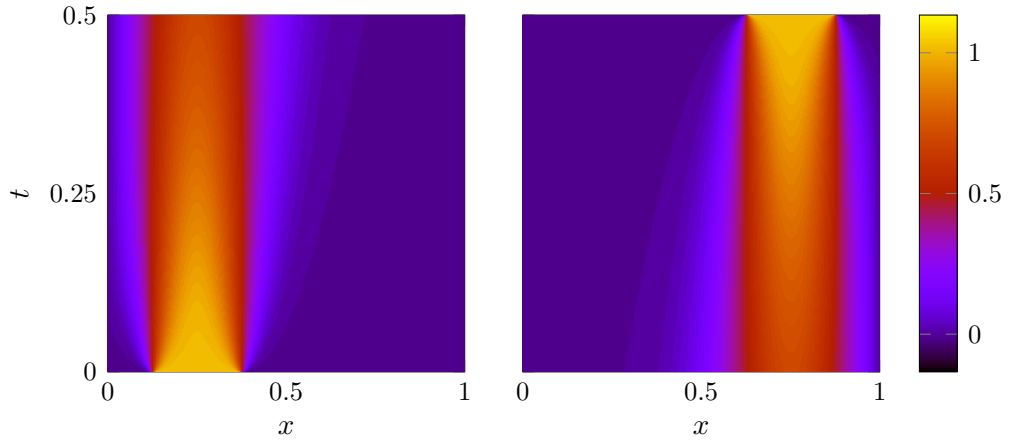



Figure 4.13.: Solution of the primal (left) and dual (right) problems.

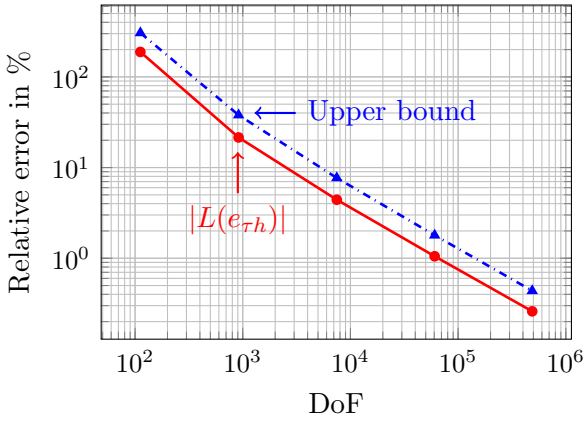



Figure 4.14.: Error in the QoI and upper bound (4.34) for uniform space-time refinements.

Figures 4.15 and 4.16 show the adapted space-time meshes and the relative errors in the QoI, respectively, when we employ $B^-_{DG}(\cdot,\cdot)$ and $B^{*-}_{DG}(\cdot,\cdot)$ as adaptive criteria. Both graphics in Figure 4.16 are the same because the problem is symmetric. Figure 4.17 shows the adapted space-time mesh after 4 iterations and the relative error in the QoI when we employ the symmetric operator (4.38). We conclude that with estimator $B^-_{DG}(\cdot,\cdot)$ or $B^{*-}_{DG}(\cdot,\cdot)$, we need 10 iterations to achieve an error of 3% while with symmetric estimator (4.38) we only need four iterations. Moreover, the space-time mesh obtained employing (4.38) is symmetric.

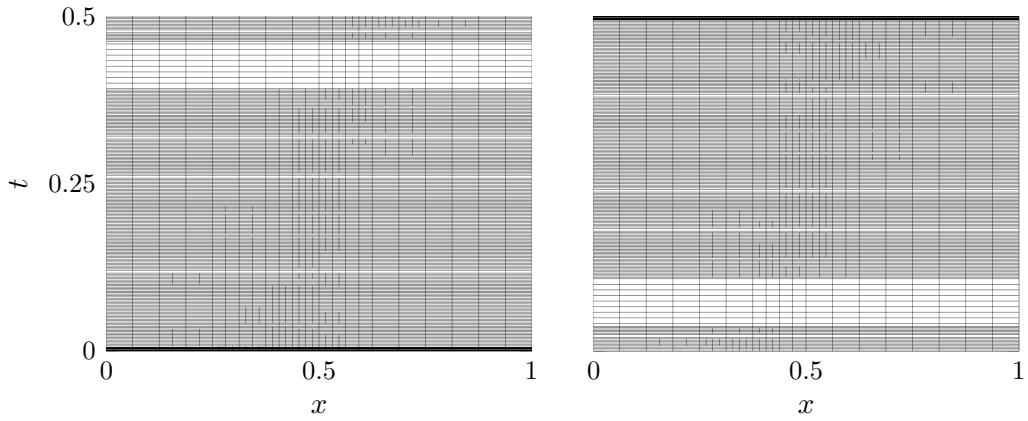



Figure 4.15.: Adapted mesh employing $B^-_{DG}(\cdot,\cdot)$ (left) and adapted mesh employing $B^{*-}_{DG}(\cdot,\cdot)$ (right).

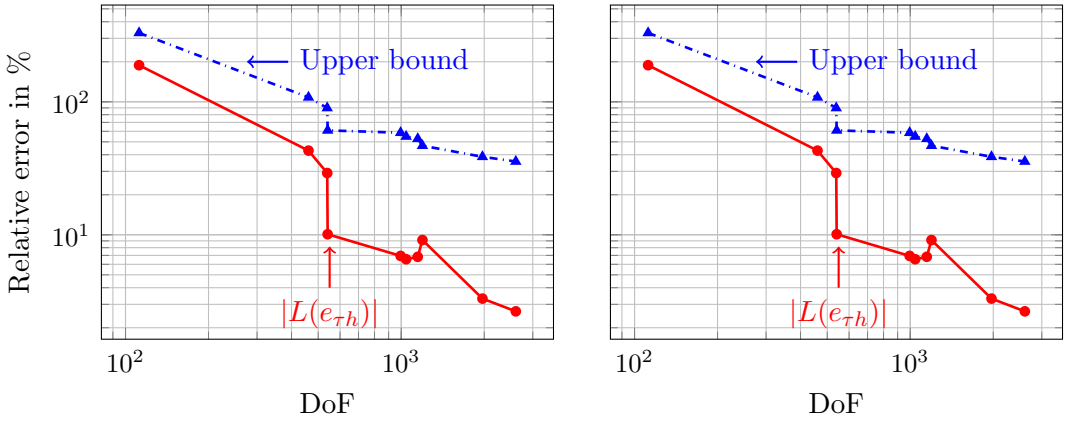



Figure 4.16.: Error in the QoI and upper bounds employing $B^{-}_{DG}(\cdot,\cdot)$ (left) and $B^{*-}_{DG}(\cdot,\cdot)$ (right).

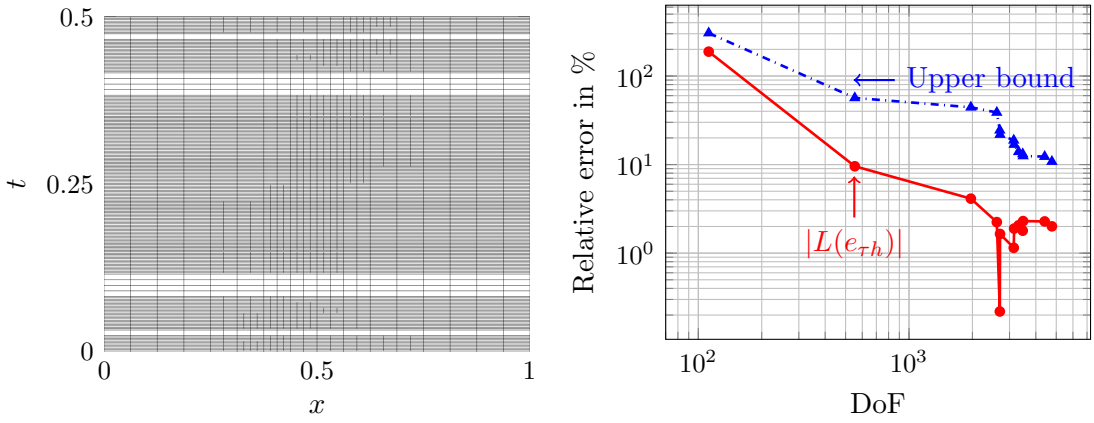



Figure 4.17.: Adapted space-time mesh (left) and error in the QoI employing symmetric estimator (4.38) (right).

Another strategy could be to select, for each space-time element, the following estimator

$$\mu_i^k := \max\{\eta_i^k, \eta_i^{*,k}\},\ \forall k = 0, \dots, m, \tag{4.39}$$

where $\eta_i^k$ are the estimators defined in (4.36) and $\eta_i^{*,k}$ are defined from the dual operator $\forall k = 1, \dots, m$

$$\eta_i^{*,k} := \left(\varepsilon_h^{k-1} - \varepsilon_h^k, e_h^{k-1}\right)_{\Omega_i^k} + \tau_k \left(\nu \nabla \varepsilon_h^k, \nabla e_h^{k-1}\right)_{\Omega_i^k} - \tau_k \left(\beta \cdot \nabla \varepsilon_h^k, e_h^{k-1}\right)_{\Omega_i^k},$$
$$\eta_i^{*,m} := \left(\varepsilon_h^m, e_h^m\right)_{\Omega_i^m}.$$

Figure 4.18 shows the adapted space-time mesh after 5 iterations and the relative error in the QoI when we employ operator (4.39). The estimator (4.39) also

yields a symmetric mesh. In this example, with estimator (4.38) we need more than $10^3$ degrees of freedom the achieve an error of 3% while with estimator (4.39) we need less than $10^3$ degrees of freedom the reach an error of 1%, although in this last case, we observe an errant behavior of the error in the QoI.

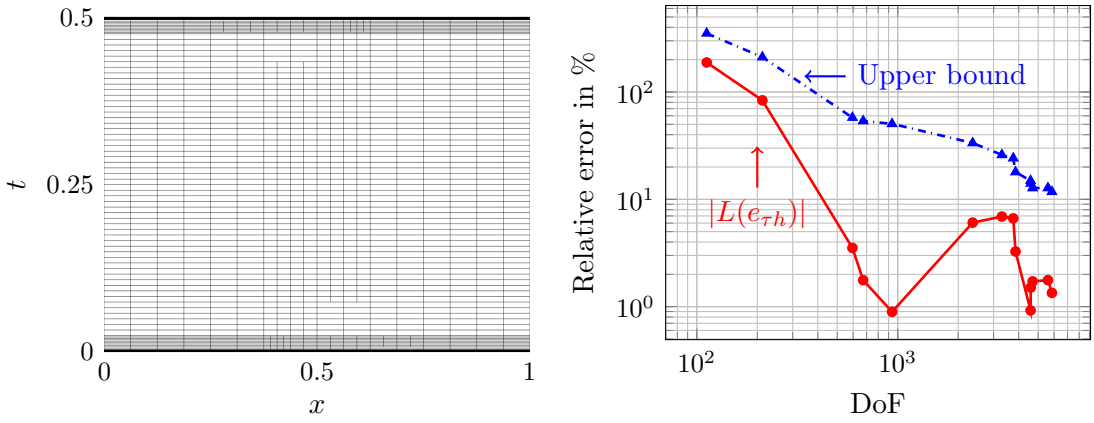



Figure 4.18.: Adapted space-time mesh (left) and error in the QoI employing symmetric estimator (4.39) (right).

It is of a great interest in this area the work developed by Parés et. al. in [116, 117], where the authors define primal and adjoint errors that are symmetric both in space and time to perform the GOA. A similar approach could be pursued on our work, and this could be a subject for future contributions.

In conclusion, employing the dPG formulation of Chapter 3, we propose an explicit-in-time adaptive algorithm that performs local goal-oriented adaptive refinements in space. The time grid is adapted locally (for the entire space) based on the CFL condition in order to ensure the stability of the method. We illustrate the performance of the proposed algorithm in one-dimensional diffusion and advection-diffusion problems.

# 5. Forward-in-time Goal-Oriented Adaptivity (GOA)

The existing goal-oriented adaptive strategies based on time-marching schemes require to solve the primal problem forwards in time, while the dual problem is solved backwards in time. This entails a significant implementation burden and additional computational cost. To overcome this problem, in this chapter, following the same argument as in [40, 41], we introduce an alternative dual problem, which we denote pseudo-dual problem. As for the primal problem, the pseudo-dual one runs forwards in time. Since we cannot ensure the Galerkin orthogonality property for this pseudo-dual problem, we employ the error representation (4.11), which is sufficient to drive the adaptive process. Then, we propose a forward-in-time adaptive process. In some applications, this approach produces outstanding results, while in others, it falls short from producing optimal refinements. For those problems, we propose a hybrid algorithm.

## 5.1. Classical goal-oriented adaptive algorithm

In this section, we solve the primal (2.11) and dual problems (4.4) selecting $r = 0$, which lead to implicit time-marching schemes. We follow the goal-oriented adaptive strategy in space for a fixed time mesh defined in [131].

As is Section 4.4, we approximate the solutions of the primal and dual problems by enriching the subspace $\mathcal{V}^r_{\tau h}$ defined in (2.10), i.e.,

$$u \simeq u_{\tau\frac{h}{2}} \in \mathcal{U}^r_{\tau\frac{h}{2}}, \quad z \simeq z_{\tau\frac{h}{2}} \in \mathcal{V}^r_{\tau\frac{h}{2}}.$$

We define the following error

$$e_{\tau\frac{h}{2}} := u_{\tau\frac{h}{2}} - u_{\tau h},$$

and we approximate the exact error as

$$e = u - u_{\tau h} \sim u_{\tau\frac{h}{2}} - u_{\tau h} = e_{\tau\frac{h}{2}}.$$

We focus on reducing the error in the Quantity of Interest (QoI) coming from the spatial discretization, i.e., $L(e_{\tau\frac{h}{2}})$ and we employ the following error representation

$$L(e_{\tau\frac{h}{2}}) = B^{+}_{DG}(e_{\tau\frac{h}{2}}, z_{\tau\frac{h}{2}}),$$

or equivalently from (2.8)

$$\begin{aligned} L(e_{\tau\frac{h}{2}}) = \sum_{k=1}^{m}\sum_{i=1}^{n_k} B_{\Omega_i^k}(e_{\tau\frac{h}{2}}, z_{\tau\frac{h}{2}}) + \left(\llbracket e_{\tau\frac{h}{2}} \rrbracket^{k-1}, z_{\tau\frac{h}{2}}(t_{k-1}^{+})\right)_{\Omega_i^k} + \\ + \sum_{i=1}^{n_0}\left(e_{\tau\frac{h}{2}}(0^{-}), z_{\tau\frac{h}{2}}(0^{-})\right)_{\Omega_i^0}, \end{aligned} \tag{5.1}$$

where $\{\Omega_i^k\}_{i=1,\dots,n_k}$, $\forall k = 0,\dots,m$ is a partition of the spatial domain $\Omega$ at $t = t_k$. Here, $(\cdot,\cdot)_{\Omega_i^k}$ denote the restriction of $(\cdot,\cdot)$ to each element $\Omega_i^k$ and $B_{\Omega_i^k}$ the restriction of $B(\cdot,\cdot)$ to each cell $\Omega_i^k \times I_k$.

Applying the triangle inequality in (5.1), we obtain the following upper bound of the error in the QoI

$$\begin{aligned} |L(e_{\tau\frac{h}{2}})| \leqslant \sum_{k=1}^{m}\sum_{i=1}^{n_k}\left|B_{\Omega_i^k}(e_{\tau\frac{h}{2}}, z_{\tau\frac{h}{2}}) + \left(\llbracket e_{\tau\frac{h}{2}} \rrbracket^{k-1}, z_{\tau\frac{h}{2}}(t_{k-1}^{+})\right)_{\Omega_i^k}\right| + \\ + \sum_{i=1}^{n_0}\left|\left(e_{\tau\frac{h}{2}}(0^{-}), z_{\tau\frac{h}{2}}(0^{-})\right)_{\Omega_i^0}\right|. \end{aligned} \tag{5.2}$$

Now, we define the error estimator of each time step as

$$Est_k := \sum_{i=0}^{n_k}\eta_i^k, \ \forall k = 0,\dots,m, \tag{5.3}$$

where

$$\begin{aligned} \eta_i^0 :&= \left(e_{\tau\frac{h}{2}}(0^{-}), z_{\tau\frac{h}{2}}(0^{-})\right)_{\Omega_i^0}, \\ \eta_i^k :&= B_{\Omega_i^k}(e_{\tau\frac{h}{2}}, z_{\tau\frac{h}{2}}) + \left(\llbracket e_{\tau\frac{h}{2}} \rrbracket^{k-1}, z_{\tau\frac{h}{2}}(t_{k-1}^{+})\right)_{\Omega_i^k}, \end{aligned} \tag{5.4}$$

$\forall i = 1,\dots,n_k$, and $\forall k = 1,\dots,m$.

Finally, Figure 5.1 illustrates a classical process to perform GOA in space for a fixed time grid. The input arguments of the classical algorithm are the time grid $\{\tau_k\}_{k=1,\dots,m}$, the spatial mesh at each time step $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, a tolerance $tol_1$ and two parameters $\theta, \lambda \in [0,1]$. We first calculate the primal solutions $u_{\tau h}$ and $u_{\tau\frac{h}{2}}$ forwards in time. Then, we compute the dual solution $z_{\tau\frac{h}{2}}$ and the estimators $Est_k$ backwards in time. For all spatial meshes satisfying $|Est_k| \geqslant \theta \cdot \max_{0\leqslant k\leqslant m}|Est_k|$,

we refine those elements in space that satisfy $|\eta_i^k| \geqslant \lambda \cdot \max_{1 \leqslant i \leqslant n_k} |\eta_i^k|$. The process ends when the relative error $QoI = 100 \cdot |L(e_{\tau \frac{h}{2}})|/|L(u_{\tau \frac{h}{2}})|$ is lower than the fixed tolerance $tol_1$.

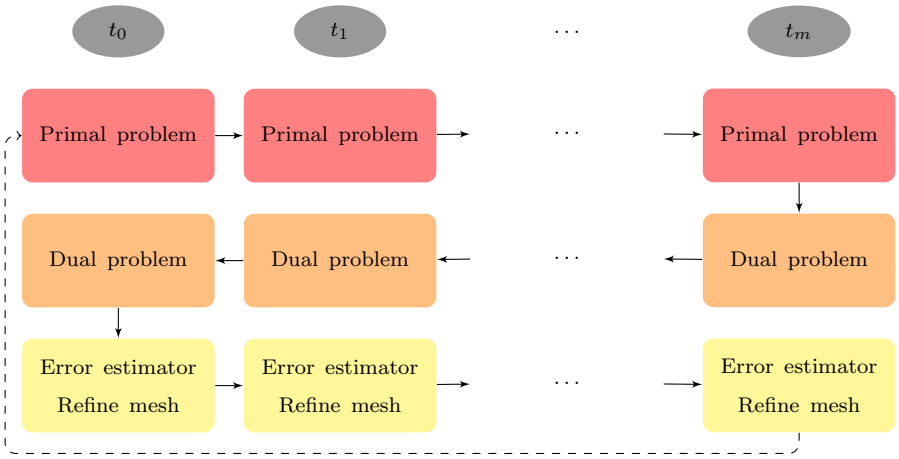



Figure 5.1.: Classical goal-oriented adaptive algorithm.

## 5.2. Pseudo-dual problem

Now, we introduce the following pseudo-dual problem

$$\begin{cases} \text{Find } \tilde{z} \in \mathcal{U} \text{ such that} \\ \tilde{B}(\tilde{z}, v_\tau) = L(v_\tau), \ \forall v_\tau \in \mathcal{V}_\tau, \end{cases} \tag{5.5}$$

where $\tilde{B}(\cdot,\cdot)$ is an alternative bilinear form and $\mathcal{V}_\tau$ the space defined in (4.7). Since the true error $e = u - u_{\tau h} \in \mathcal{V}_\tau$, we obtain the following error representation,

$$L(e) = \tilde{B}(\tilde{z}, e). \tag{5.6}$$

For example, if we select $\tilde{B}(\tilde{z}, v) = B_{DG}^+(\tilde{z}, v)$ (being $B_{DG}^+(\cdot,\cdot)$ the bilinear form defined in (2.8)) as $\tilde{z}$, is a continuous function in time, we have that

$$B(\tilde{z}, v_\tau) + (\tilde{z}(0), v_\tau(0)) = \int_I \langle g, v_\tau \rangle \, dt + (z_T, v_\tau(T)), \ \forall v_\tau \in \mathcal{V}_\tau.$$

Now, as explained in [11], we can express the right-hand-side of the previous equation as

$$B(\tilde{z}, v_\tau) + (\tilde{z}(0), v_\tau(0)) = \int_I \langle g, v_\tau \rangle \, dt + \int_I (\delta(t-T) z_T, v_\tau) \, dt, \ \forall v_\tau \in \mathcal{V}_\tau, \tag{5.7}$$

where $\delta(t-T)$ is a Dirac delta distribution. Therefore, following the same argument as in Section 4.1, we conclude that the initial condition of the pseudo-dual problem is zero and both functions $g$ and $z_T$ are part of the source.

We can solve problem (5.5) forward in time using a time-marching scheme. However, as we show in the numerical results, the pseudo-dual problem is only properly defined for particular problems.

## 5.3. Forward-in-time goal-oriented adaptive algorithm

We now describe a goal-oriented adaptive strategy based on the pseudo-dual problem (5.5) when $\tilde{B}(\tilde{z},v) = B^{+}_{DG}(\tilde{z},v)$ that runs forwards is time.

From (5.6), we have

$$L(e_{\tau\frac{h}{2}}) = B^{+}_{DG}(\tilde{z}_{\tau\frac{h}{2}}, e_{\tau\frac{h}{2}}),$$

and obtain an upper bound of the error in the QoI

$$\begin{aligned} |L(e_{\tau\frac{h}{2}})| \leqslant \sum_{k=1}^{m}\sum_{i=1}^{n_k} \left| B_{\Omega_i^k}(\tilde{z}_{\tau\frac{h}{2}}, e_{\tau\frac{h}{2}}) + \left( [\![ \tilde{z}_{\tau\frac{h}{2}} ]\!]^{k-1}, e_{\tau\frac{h}{2}}(t^{+}_{k-1}) \right)_{\Omega_i^k} \right| + \\ + \sum_{i=1}^{n_0} \left| \left( \tilde{z}_{\tau\frac{h}{2}}(0^{-}), e_{\tau\frac{h}{2}}(0^{-}) \right)_{\Omega_i^0} \right|, \end{aligned} \tag{5.8}$$

We construct the estimators following a similar technique to the one we describe for (5.3) and (5.4). The algorithm we propose describes a forward-in-time goal-oriented adaptive process for the spatial error for a fixed time grid. As before, the input arguments are the time grid $\{\tau_k\}_{k=1,\dots,m}$, the spatial mesh at each time step $\{\mathcal{M}_h^k\}_{k=0,\dots,m}$, a tolerance $tol_2$, a parameter $\lambda \in [0,1]$ and the maximum number of iterations per time step. We calculate the solutions $u_{\tau h}$, $u_{\tau\frac{h}{2}}$, $z_{\tau\frac{h}{2}}$ and the estimators $Est_k$ at each time step. Then, if $|Est_k|$ is greater than a tolerance $tol_2$, we refine the elements in space satisfying $|\eta_i^k| \geqslant \lambda \cdot \max\limits_{1\leqslant i\leqslant n_k} |\eta_i^k|$. Figure 5.2 illustrates the proposed adaptive algorithm.

Although the computational time in the proposed forward-in-time adaptive algorithm is similar to the classical one, there is a considerable saving in memory storage. In this algorithm, we do not need to store the solutions of the primal and dual problems at *all* times to perform the adaptive process. The adaptivity is performed as the solutions are calculated. However, as we show in the numerical results, the pseudo dual problem we proposed in (5.5) is inadequate to solve certain problems. In those situations, we propose a hybrid algorithm in the next section.

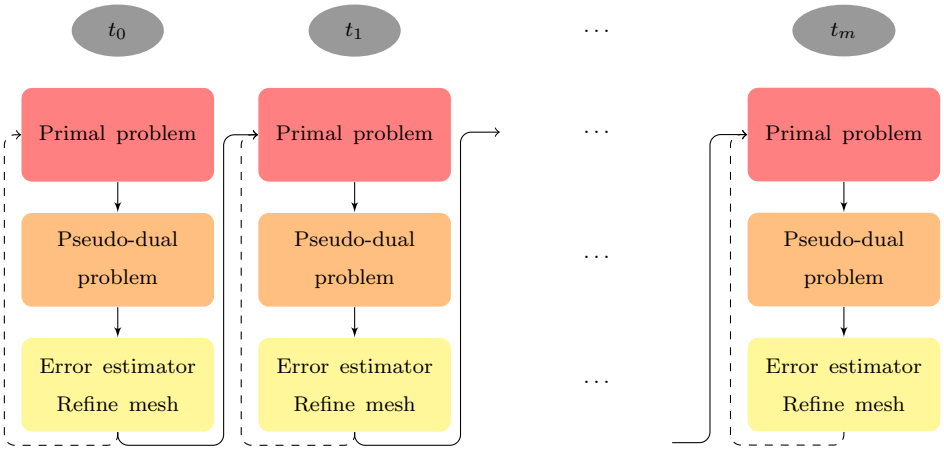



Figure 5.2.: Proposed forward-in-time goal-oriented adaptive algorithm.

## 5.4. Hybrid algorithm

We select the pseudo-dual problem (5.5) with

$$\tilde{B}(\tilde{z}, v_\tau) = B^+_{DG}(\tilde{z}, v_\tau) - (z_{\tau\frac{h}{2}}(0), v_\tau(0)), \tag{5.9}$$

where $z_{\tau\frac{h}{2}}(0)$ is the approximated solution of the classical dual problem (4.4) at $t = 0$. Selecting (5.9), formula (5.7) becomes

$$B(\tilde{z}, v_\tau) + (\tilde{z}(0), v_\tau(0)) = \int_I \langle g, v_\tau \rangle \, dt + \int_I (\delta(t-T) z_T, v_\tau) \, dt + (z_{\tau\frac{h}{2}}(0), v_\tau(0)),$$

and therefore, the initial condition of the pseudo-dual problem is $z_{\tau\frac{h}{2}}(0)$.

Now, from (5.9) we have that

$$\begin{aligned} |L(e_{\tau\frac{h}{2}})| \leqslant \sum_{k=1}^{m} \sum_{i=1}^{n_k} \left| B_{\Omega_i^k}(\tilde{z}_{\tau\frac{h}{2}}, e_{\tau\frac{h}{2}}) + \left( [[\tilde{z}_{\tau\frac{h}{2}}]]^{k-1}, e_{\tau\frac{h}{2}}(t^+_{k-1}) \right)_{\Omega_i^k} \right| + \\ + \sum_{i=1}^{n_0} \left| \left( \tilde{z}_{\tau\frac{h}{2}}(0^-), e_{\tau\frac{h}{2}}(0^-) \right)_{\Omega_i^0} - \left( z_{\tau\frac{h}{2}}(0^-), e_{\tau\frac{h}{2}}(0^-) \right)_{\Omega_i^0} \right|. \end{aligned} \tag{5.10}$$

However, $\tilde{z}_{\tau\frac{h}{2}}(0^-) = z_{\tau\frac{h}{2}}(0^-)$ from the definition of (5.9), so the last term of (5.10) is zero and we do not have an error indicator for the first mesh.

Therefore, we define the following upper bound of the error in the QoI that is larger than (5.10) but includes an estimator for the first mesh in time as in

Sections 5.1 and 5.3

$$|L(e_{\tau\frac{h}{2}})| \leqslant \sum_{k=1}^{m}\sum_{i=1}^{n_k}\left|B_{\Omega_i^k}(\tilde{z}_{\tau\frac{h}{2}}, e_{\tau\frac{h}{2}}) + \left([[\tilde{z}_{\tau\frac{h}{2}}]]^{k-1}, e_{\tau\frac{h}{2}}(t_{k-1}^{+})\right)_{\Omega_i^k}\right| + \\ + \sum_{i=1}^{n_0}\left|\left(\tilde{z}_{\tau\frac{h}{2}}(0^{-}), e_{\tau\frac{h}{2}}(0^{-})\right)_{\Omega_i^0}\right|. \tag{5.11}$$

As in Section 5.1, from (5.11) we define the error estimators $Est_k$ and $\eta_i^k$, $\forall i = 1, \ldots, n_k$ and $\forall k = 0, \ldots, m$. In this hybrid algorithm, we first solve the classical dual problem (4.4) in the fine mesh backwards in time. Then, we follow the same strategy defined in Section 5.3 employing $z_{\tau\frac{h}{2}}(0)$ as the initial condition for the pseudo-dual problem. Therefore, we need to solve the dual problem backwards in time once and then we can perform the entire goal-oriented adaptive process forwards in time. In this algorithm, although we solve the classical dual problem once, we only need to store the solution in the first time step to start the adaptive process forwards in time as in Section 5.3. Figure 5.3 illustrates the proposed hybrid adaptive algorithm.

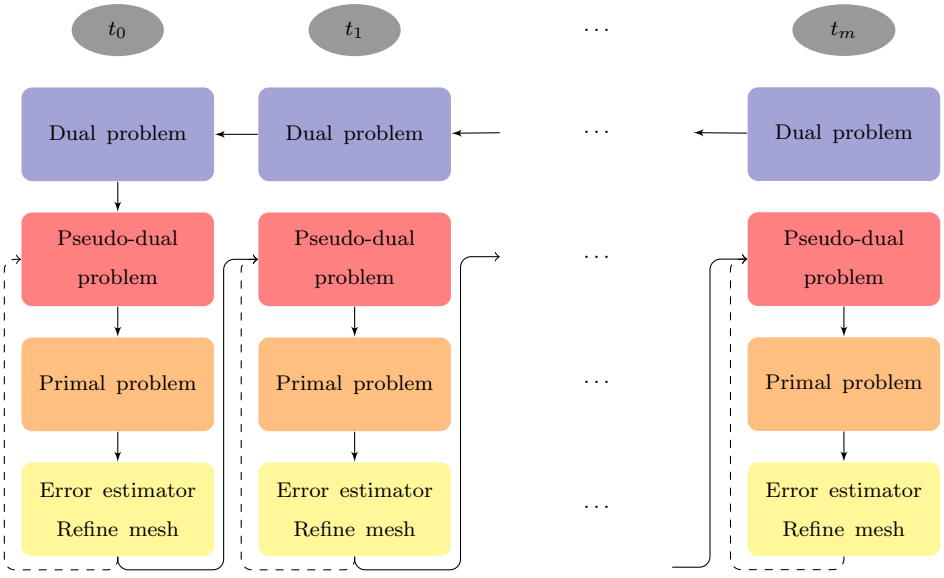



Figure 5.3.: Proposed hybrid goal-oriented adaptive algorithm.

## 5.5. Numerical results

### 5.5.1. Diffusion problem

We consider problem (2.1) with $d = 1$, $\Omega = [0, 1]$, $T = 1$, $f(x, t) = (1 + \pi^2 t)\sin(\pi x)$, $u(0) = 0$, $\beta = 0$ and a discontinuous diffusion coefficient

$$\nu(x) = \begin{cases} 10, & x \in [0.25, 0.75], \\ 0.01, & \text{elsewhere.} \end{cases}$$

We also consider the following output functional

$$L(u) = \int_I \int_{\Omega_0} u(x, t) dx dt,$$

where $\Omega_0 = (0, 0.25) \cup (0.75, 1)$ is a subdomain of $\Omega$. From (4.3), we have that the final condition of the dual problem is null and the source term is a function whose value is 1 in $\Omega_0$ and vanishes outside $\Omega_0$.

For the discretization, we employ constant-in-time basis functions ($r = 0$) in (2.11) and (4.4), and linear functions in space. We set 100 time steps and in space, we start the adaptive process with a coarse mesh consisting of 8 elements.

Figure 5.4 shows the solution of the primal problem (2.2) and Figure 5.5 shows the solution of the dual problems (4.2) and (5.5). We can see that the problem (5.5) is the same as (4.2) but running forward in time.

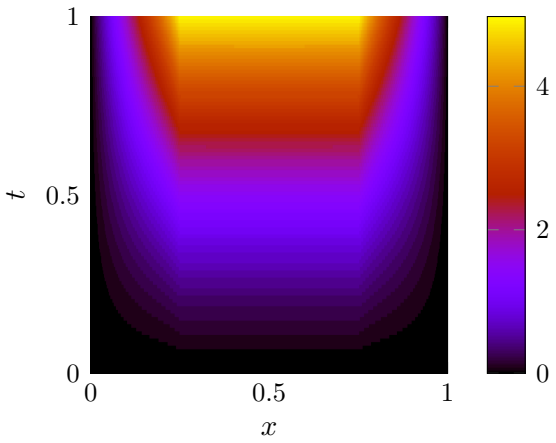



Figure 5.4.: Solution of the primal problem.

Now, we perform GOA employing both the classical strategy and the proposed forward-in-time process. Figure 5.7 shows the estimation and the upper bound of the error in the QoI using the classical approach and setting $tol_1 = 10^{-3}$

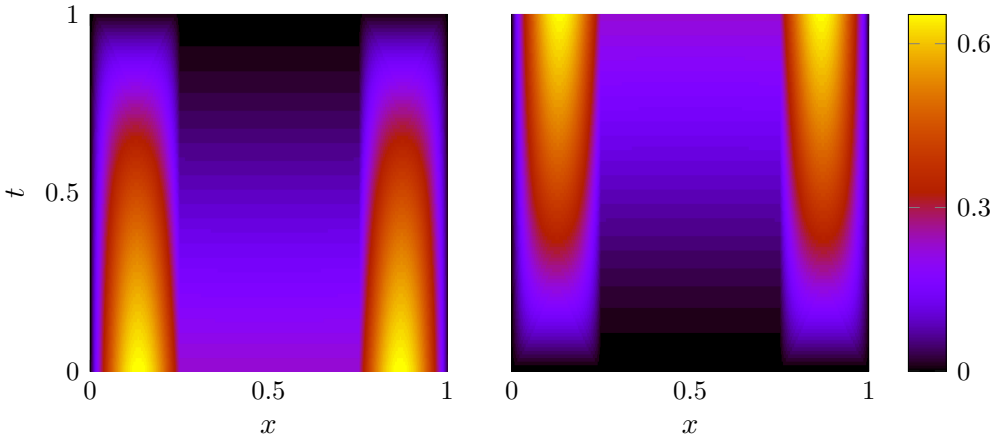



Figure 5.5.: Solution of the dual (left) and pseudo-dual (right) problems.

and $\theta = \lambda = 0.25$. Figure 5.7 shows the error and the upper bound for the proposed algorithm when we set $tol_2 = tol_1/100$, $\lambda = 0.25$ and a maximum of seven iterations per time step. Finally, Figure 5.8 shows the adapted grids using both algorithms. We conclude that both grids are similar but with our algorithm we obtain more uniform refinements in the area where the diffusion term is discontinuous.

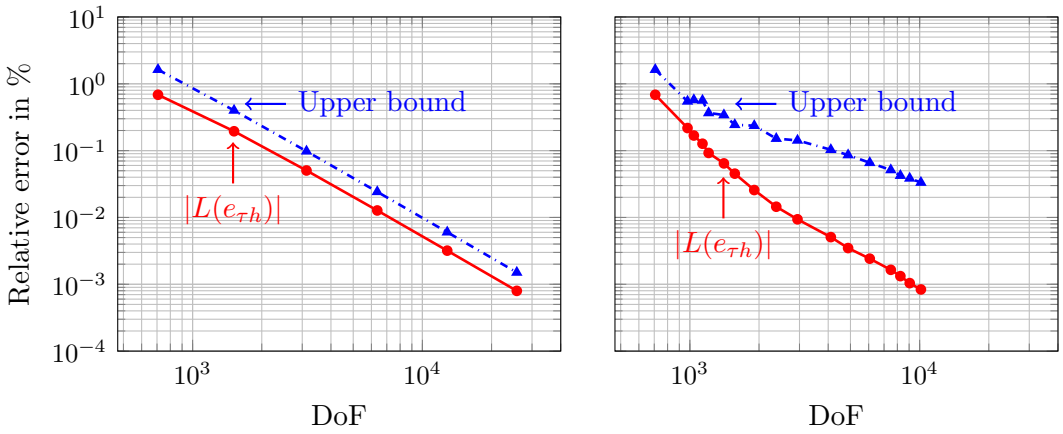



Figure 5.6.: Error in the QoI and upper bound (5.2) for uniform refinements in space (left) and using the classical algorithm (right).

We conclude that, for this problem, the forward-in-time adaptive algorithm performs similarly to the classical algorithm. We achieve a relative error of $10^{-3}$ with $10^4$ degrees of freedom in both cases. However, the proposed forward-in-time adaptive algorithm is computationally cheaper than the classical one as it per-

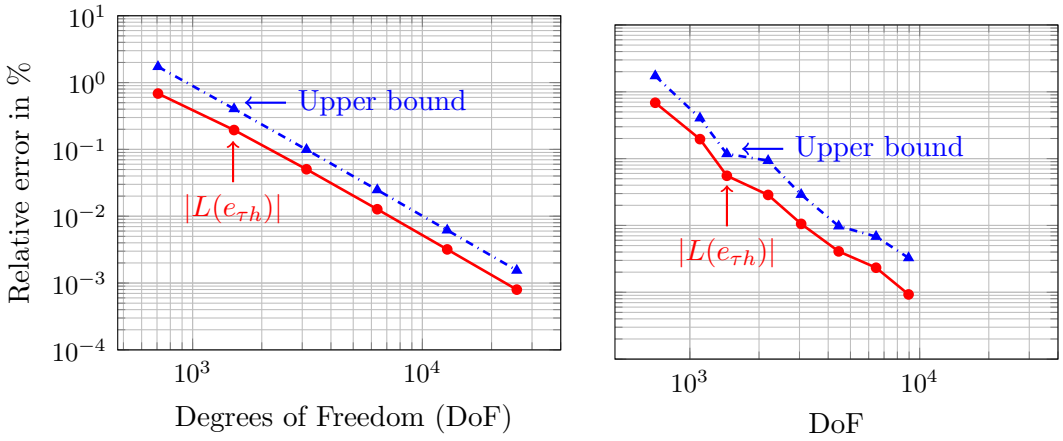



Figure 5.7.: Error in the QoI and upper bound (5.8) for uniform refinements in space (left) and using the forward-in-time algorithm (right).

forms the adaptivity while both problems, primal and dual, are solved forwards in time. In that way, it is unnecessary to store the adjoint solution for all time steps, and one can only save the previous and current time steps solutions, maximizing memory savings. This also simplifies data structures and implementation. Also, it is expected that this algorithm will minimize the number of adaptive iterations, as confirmed via numerical results.

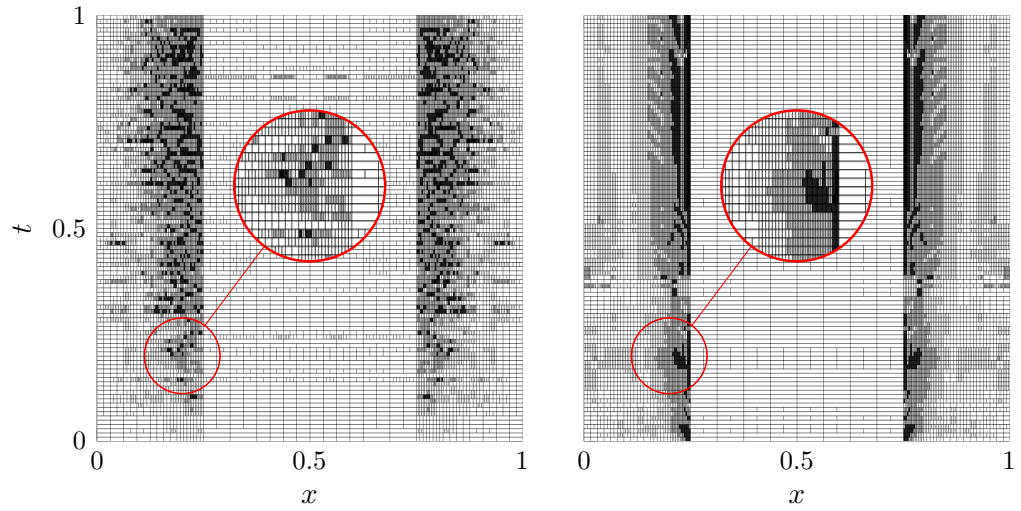



Figure 5.8.: Adapted grids employing the classical algorithm (left) and the forward-in-time algorithm (right).

### 5.5.2. Advection-diffusion problem

Let $d = 1$, $\Omega = [0, 1]$, $T = 0.25$, $\nu = 0.025$, $\beta = 2.5$ $f(x, t) = 0$ and a discontinuous initial condition

$$u_0(x) = \begin{cases} 1, & x \in [0.125, 0.375], \\ 0, & \text{elsewhere.} \end{cases}$$

As in Section 4.6.2, the initial condition is propagated due to the positive advection coefficient and a boundary layer is formed at the final time steps in the right endpoint on the spatial domain. We seek to reduce the error in the boundary layer, therefore we consider an output functional of the form

$$L(u) = \int_{I_0} \int_{\Omega_0} u(x, t) dx dt,$$

where $I_0 \times \Omega_0 = (0.75, 1) \times (0.2, 0.25) \subset I \times \Omega$.

As before, we set 100 time steps and in space and we start the adaptive process with a coarse mesh composed of 8 elements. Figure 5.9 shows the solution of the primal problem (2.2) and Figure 5.10 shows the solution of the dual problem (4.2) and the pseudo-dual problem (5.5). In this case, the solution of the pseudo-dual problem is zero in the entire space-time domain except in the QoI.

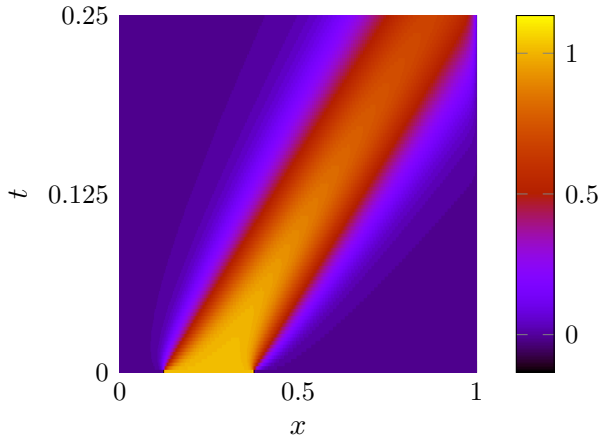



Figure 5.9.: Solution of the primal problem.

Figure 5.11 shows the error in the QoI and the upper bound using the classical algorithm when we set $tol_1 = 10^{-1}$ and $\theta = \lambda = 0.2$. Figure 5.12 shows the error and the upper bound employing the proposed algorithm setting $tol_2 = 0$, $\lambda = 0.2$ and a maximum of seven iterations per time steps. Finally, Figure 5.13 shows the adapted grids using both processes.

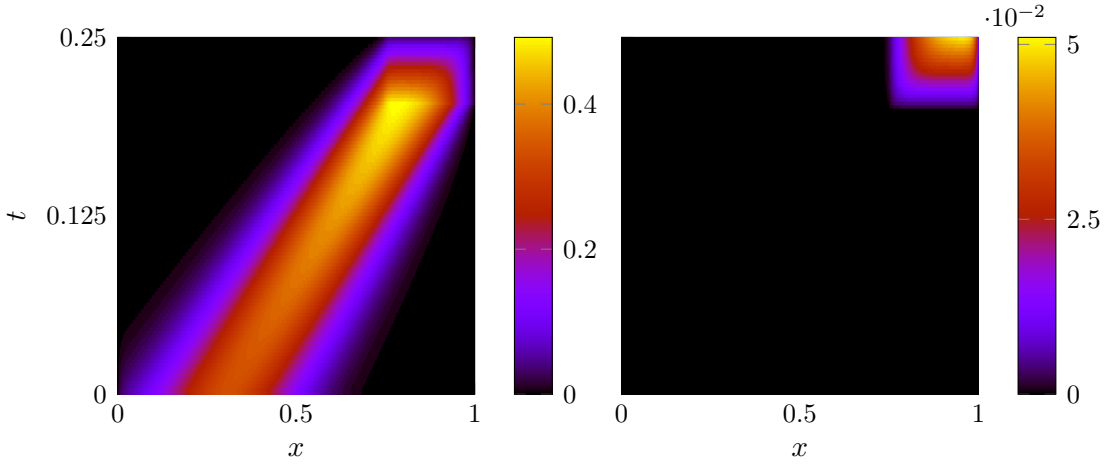



Figure 5.10.: Solution of the dual (left) and pseudo-dual (right) problems.

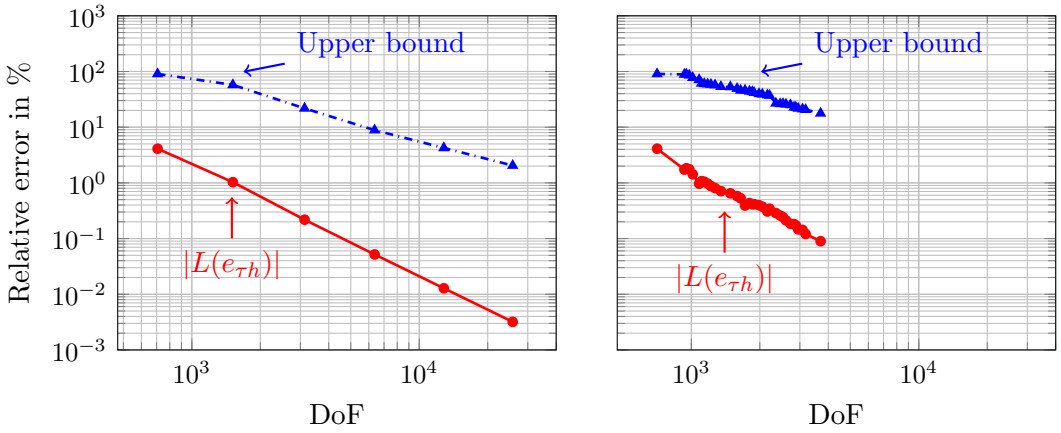



Figure 5.11.: Error in the QoI and upper bound (5.2) for uniform refinements in space (left) and using the classical algorithm (right).

We conclude that, for this kind of problems, the proposed forward-in-time adaptive process performs poorly because the algorithm only produces refinements within the support of the output functional. By construction, this algorithm ignores the propagation of the initial condition of the primal problem due to the advection term. As a partial remedy to this limitation of our forward-in-time algorithm, in the next section, we propose a hybrid algorithm that provides better results for advection-diffusion problems.

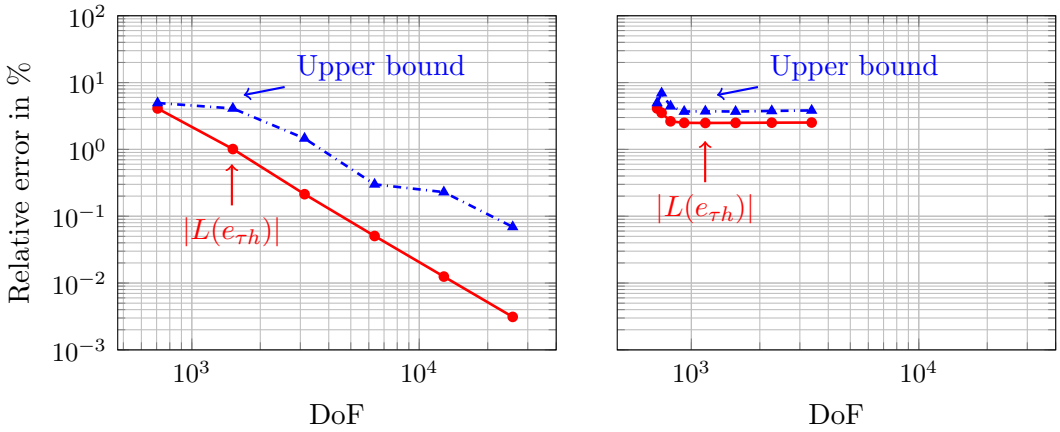



Figure 5.12.: Error in the QoI and upper bound (5.8) for uniform refinements in space (left) and using the forward-in-time algorithm (right).

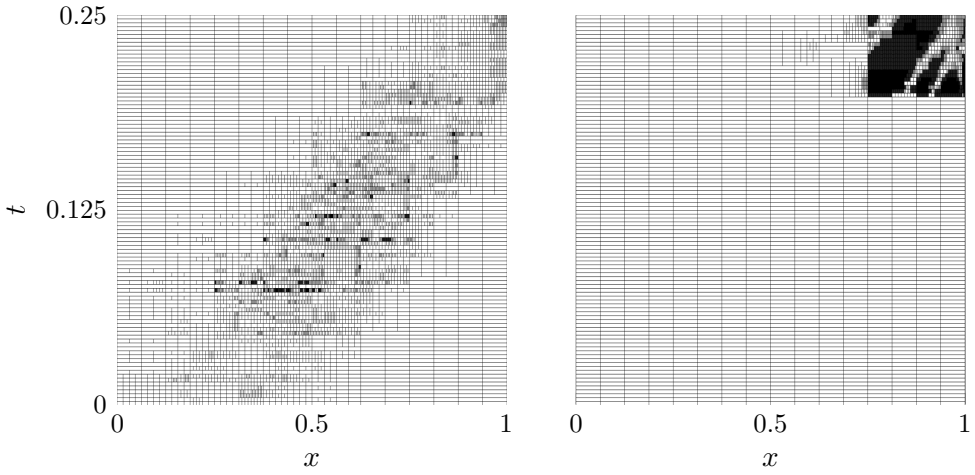



Figure 5.13.: Adapted grids employing the classical algorithm (left) and the forward-in-time algorithm (right).

### 5.5.3. Hybrid algorithm for advection-diffusion problems

We consider the same example as in Section 5.5.2 and we employ the hybrid algorithm described in Section 5.4 Figure 5.14 shows the solution of the pseudo-dual problem.

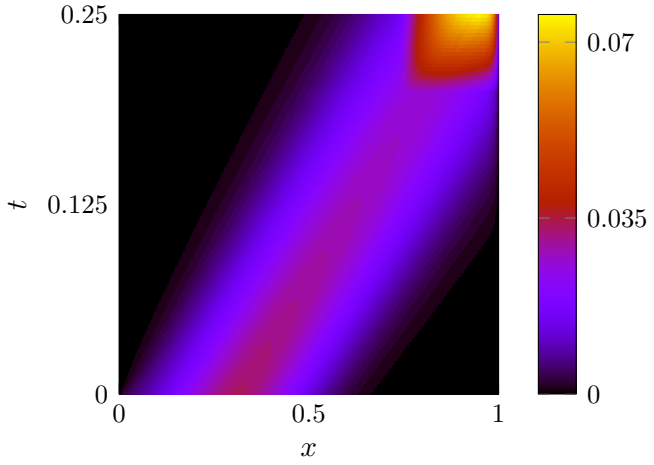



Figure 5.14.: Solution of the pseudo-dual problem.

For the hybrid algorithm, we set $tol_2 = 0$, $\lambda = 0.2$ and a maximum of seven iterations per time step. Figure 5.15 shows the error in the QoI and its upper bound, while Figure 5.16 exhibits the adapted grid employing the proposed hybrid algorithm. We conclude that the hybrid algorithm not only refines the mesh where the support of the output functional is localized, but it also captures the path of the advection.

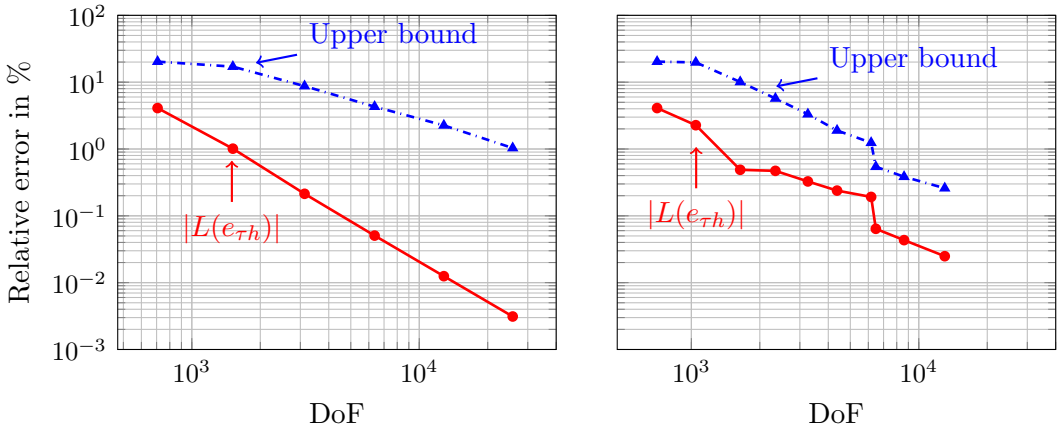



Figure 5.15.: Error in the QoI and upper bound (5.11) for uniform refinements in space (left) and employing the hybrid algorithm (right).

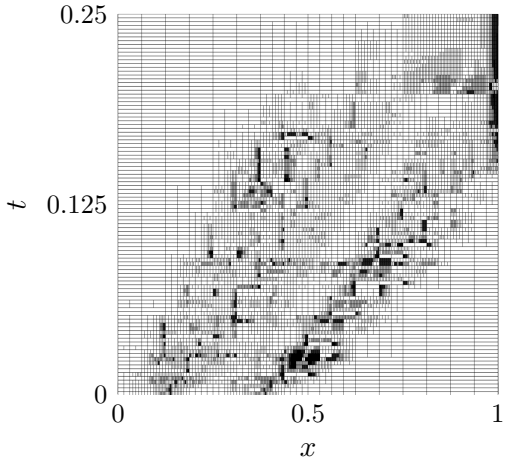



Figure 5.16.: Adapted grid employing the proposed hybrid algorithm.

In conclusion, employing a pseudo-dual problem running forwards in time, we propose a forward-in-time goal-oriented adaptive algorithm for the linear advection-diffusion equation. We compare the proposed algorithm with the classical one via one-dimensional numerical results. We conclude that our algorithm only performs properly for diffusion problems where the output functional has support along the whole time interval. For advection-diffusion problems, we propose a hybrid algorithm, which solves the classical dual problem backwards in time once and then, it performs the adaptive process forwards in time. In both algorithms, we only need to store the solutions of the primal and dual problems at one time step to perform the adaptivity.

# 6. Alternative bounds for wave propagation problems

In this chapter, we apply the pseudo-dual error representation of Darrigrand et al. [40] to the 1D+time wave equation. The main objective is to obtain sharper upper bounds of the error representation in the goal-oriented approach. First, we describe the continuous-in-time Petrov-Galerkin (cPG) formulation and the discretization of the model problem that we study throughout this chapter. Then, we describe the classical goal-oriented error representation and we introduce an alternative representation of the error in the Quantity of Interest (QoI). Finally, we describe the two refinement algorithms we employ to compare the upper bounds of both representations.

## 6.1. Model problem, variational formulation and discretization

Let $\Omega = (a, b) \subset \mathbb{R}$ and $I = (0, T] \subset \mathbb{R}$. We consider the following *one-dimensional wave equation*

$$\begin{cases} u_{tt} - (\alpha u_x)_x = f & \text{in } \Omega \times I, \\ u(a,t) = u(b,t) = 0 & \text{in } I, \\ u(x,0) = u_0(x) & \text{in } \Omega, \\ u_t(x,0) = v_0(x) & \text{in } \Omega, \end{cases} \tag{6.1}$$

where the speed of propagation of the wave $\sqrt{\alpha(x)}$, the source term $f(x,t)$, the initial position $u_0(x)$ and the initial speed $v_0(x)$ are given data that define the problem. We denote by $u_x$ and $u_t$ the partial derivatives of the solution $u(x,t)$ with respect to the spatial variable $x$ and the temporal variable $t$, respectively.

We consider the spaces $V$, $V'$, $\mathcal{V}$ and $\mathcal{V}'$, defined in Section 2.1. We denote by

$$H := L^2(\Omega), \;\; \mathcal{H} := L^2(I; H),$$

and we assume that $f(x,t) \in \mathcal{V}'$, $u_0(x) \in V$ and $v_0(x) \in H$. For simplicity, we will consider homogeneous Dirichlet conditions on the spatial boundaries.

We perform the change of variables $v = u_t$, so system (6.1) becomes

$$\begin{cases} u_t = v & \text{in } \Omega \times I, \\ v_t - (\alpha u_x)_x = f & \text{in } \Omega \times I, \\ u(a,t) = u(b,t) = 0 & \text{in } I, \\ u(x,0) = u_0(x) & \text{in } \Omega, \\ v(x,0) = v_0(x) & \text{in } \Omega, \end{cases} \tag{6.2}$$

We know from [9, 96] that the weak solution $\{u, v\}$ of problem (6.2) belongs to $\mathscr{U} \times \mathscr{V}$, where

$$\begin{aligned} \mathscr{U} &:= \{u \in \mathcal{V} \mid u_t \in \mathcal{H},\ u_{tt} \in \mathcal{V}'\}, \\ \mathscr{V} &:= \{v \in \mathcal{H} \mid v_t \in \mathcal{V}'\}, \end{aligned}$$

and it satisfies $u(x,0) = u_0(x)$, $v(x,0) = v_0(x)$ and

$$\begin{aligned} (u_t, w) - (v, w) &= 0,\ \forall w \in H, \\ \langle v_t, z\rangle + (\alpha u_x, z_x) &= \langle f, z\rangle,\ \forall z \in V. \end{aligned} \tag{6.3}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $V$ and $V'$, and $(\cdot, \cdot)$ denotes the inner product in $H$.

Using the spaces defined above, the weak solution $\{u, v\}$ is globally continuous in time. Now, we select $\mathcal{H}$ and $\mathcal{V}$ as test spaces. Integrating in time the expressions in (6.3) and imposing the initial conditions in weak form, we conclude that (6.3) is equivalent to

$$\begin{aligned} \int_I (u_t, w)dt - \int_I (v, w)dt &= 0,\ \forall w \in \mathcal{H}, \\ (u(0), \hat{w}) &= (u_0, \hat{w}),\ \forall \hat{w} \in H, \\ \int_I \langle v_t, z\rangle\, dt + \int_I (\alpha u_x, z_x)dt &= \int_I \langle f, z\rangle\, dt,\ \forall z \in \mathcal{V}, \\ (v(0), \hat{z}) &= (v_0, \hat{z}),\ \forall \hat{z} \in H, \end{aligned} \tag{6.4}$$

where $u(0) := u(x,0)$ and $v(0) := v(x,0)$.

We denote by $U := \{u, v\} \in \mathscr{U} \times \mathscr{V}$, $Z := \{w, z\} \in \mathcal{H} \times \mathcal{V}$. Then, employing the time partition defined in (2.4) and the semidiscrete spaces in (2.17), we consider the following semidiscrete cPG formulation of (6.4)

$$\begin{cases} \text{Find } U_\tau = \{u_\tau, v_\tau\} \in \mathfrak{U}_\tau^r \times \mathfrak{U}_\tau^r \text{ such that} \\ \mathcal{B}_{CG}(U_\tau, Z_\tau) = \mathcal{F}(Z_\tau),\ \forall Z_\tau = \{w_\tau, z_\tau\} \in \mathcal{V}_\tau^{r-1} \times \mathcal{V}_\tau^{r-1}, \end{cases} \tag{6.5}$$

where we write the expressions in (6.4) in a compact form by adding the equations and we define

$$\begin{aligned}\mathcal{B}_{CG}(U,Z) :&= \int_I (u_t, w)dt - \int_I (v, w)dt + (u(0), w(0)) \\ &+ \int_I \langle v_t, z\rangle\, dt + \int_I (\alpha u_x, z_x)dt + (v(0), z(0)), \\ \mathcal{F}(Z) :&= (u_0, w(0)) + \int_I \langle f, z\rangle\, dt + (v_0, z(0)).\end{aligned}$$

Now, we perform a partition of the spatial interval $\Omega = (a, b)$ into $n$ subintervals as follows

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,$$

and we denote by $\Omega_i = (x_{i-1}, x_i)$, $h_i = x_i - x_{i-1}$, $\forall i = 1, \ldots, n$ and $h = \max\limits_{1\leqslant i\leqslant n} h_i$. Related to this partition, we denote by $V_h$ the finite-dimensional subspace of $V$ generated by the continuous, piecewise linear functions defined over each subinterval.

In order to fully discretize problem (6.5), we select the following discrete spaces [9]

$$\begin{aligned}\mathfrak{U}^r_{\tau h} &:= \{u \in C(\bar{I}; V_h) \mid u_{|_{I_k}} \in P_r(\bar{I}_k; V_h),\ \forall k = 1, \ldots, m\}, \\ \mathcal{V}^{r-1}_{\tau h} &:= \{u \in L^2(I; V_h) \mid u_{|_{(t_{k-1}, t_k]}} \in P_{r-1}((t_{k-1}, t_k]; V_h),\ \forall k = 1, \ldots, m,\ u(0) \in V_h\},\end{aligned}$$

where we avoid the use of dynamic meshes in space as in previous chapters. In particular, in (6.5), we select $r = 1$. Therefore, the functions in $\mathfrak{U}^r_{\tau h}$ are piecewise linear in space and time and globally continuous, whereas functions in $\mathcal{V}^{r-1}_{\tau h}$ are continuous piecewise linear in space and discontinuous piecewise constant in time.

Finally, the fully discrete cPG formulation becomes

$$\begin{cases}\text{Find } U_{\tau h} = \{u_{\tau h}, v_{\tau h}\} \in \mathfrak{U}^r_{\tau h} \times \mathfrak{U}^r_{\tau h} \text{ such that} \\ \mathcal{B}_{CG}(U_{\tau h}, Z_{\tau h}) = \mathcal{F}(Z_{\tau h}),\ \forall Z_{\tau h} = \{w_{\tau h}, z_{\tau h}\} \in \mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h}.\end{cases} \tag{6.6}$$

## 6.2. Dual problem and error representations

We consider output functionals $\mathcal{L} : \mathscr{U} \times \mathscr{V} \longrightarrow \mathbb{R}$ of the form

$$\mathcal{L}(U) = \mathcal{L}_0(u) + \mathcal{L}_1(v),$$

with $\mathcal{L}_0 : \mathscr{U} \longrightarrow \mathbb{R}$ and $\mathcal{L}_1 : \mathscr{V} \longrightarrow \mathbb{R}$ are defined as

$$\mathcal{L}_0(u) = \int_I \langle u, g\rangle\, dt + (u(T), w_{_T}),\ \ \mathcal{L}_1(v) = (v(T), z_{_T}),$$

where $g \in \mathcal{V}'$, $w_{_T} \in H$ and $z_{_T} \in V$ are given functions.

Then, we introduce the following dual problem

$$\begin{aligned} -\int_I \langle u, w_t \rangle \, dt + \int_I (u_x, \alpha z_x) dt &= \int_I \langle u, g \rangle \, dt, \ \forall u \in \mathcal{V}, \\ (\hat{u}, w(T)) &= (\hat{u}, w_{_T}), \ \forall \hat{u} \in H, \\ -\int_I (v, z_t) dt - \int_I (v, w) dt &= 0, \ \forall v \in \mathcal{H}, \\ (\hat{v}, z(T)) &= (\hat{v}, z_{_T}), \ \forall \hat{v} \in H, \end{aligned} \tag{6.7}$$

and the corresponding strong formulation of dual problem (6.7) is

$$\begin{cases} z_t = -w & \text{in } \Omega \times I, \\ -w_t - (\alpha z_x)_x = g & \text{in } \Omega \times I, \\ z(a,t) = z(b,t) = 0 & \text{in } I, \\ z(x,T) = z_T(x) & \text{in } \Omega, \\ w(x,0) = w_T(x) & \text{in } \Omega. \end{cases} \tag{6.8}$$

We conclude from (6.8) that the dual problem is also a wave propagation problem, but running backwards in time.

## 6.2.1. Classical error representation

In order to provide an error representation for the traditional goal-oriented approach, we consider the continuous and discrete primal problems

$$\begin{cases} \text{Find } U^* = \{u^*, v^*\} \in \mathscr{U} \times \mathscr{V} \text{ and } U^*_{\tau h} = \{u^*_{\tau h}, v^*_{\tau h}\} \in \mathfrak{U}^r_{\tau h} \times \mathfrak{U}^r_{\tau h} \\ \text{such that} \\ \mathcal{B}_{_{CG}}(U^*, Z) = \mathcal{F}(Z), \ \forall Z = \{w, z\} \in \mathcal{V}_\tau \times \mathcal{V}_\tau, & \text{(6.9a)} \\ \mathcal{B}_{_{CG}}(U^*_{\tau h}, Z_{\tau h}) = \mathcal{F}(Z_{\tau h}), \ \forall Z_{\tau h} = \{w_{\tau h}, z_{\tau h}\} \in \mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h}, & \text{(6.9b)} \end{cases}$$

and their duals

$$\begin{cases} \text{Find } Z^* = \{w^*, z^*\} \in \mathcal{V}_\tau \times \mathcal{V}_\tau \text{ and } Z^*_{\tau h} = \{w^*_{\tau h}, z^*_{\tau h}\} \in \mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h} \\ \text{such that} \\ \mathcal{B}_{_{CG}}(U, Z^*) = \mathcal{L}(U), \ \forall U = \{u, v\} \in \mathscr{U} \times \mathscr{V}, & \text{(6.10a)} \\ \mathcal{B}_{_{CG}}(U_{\tau h}, Z^*_{\tau h}) = \mathcal{L}(U_{\tau h}), \ \forall U_{\tau h} = \{u_{\tau h}, v_{\tau h}\} \in \mathfrak{U}^r_{\tau h} \times \mathfrak{U}^r_{\tau h}. & \text{(6.10b)} \end{cases}$$

Here, $\mathcal{V}_\tau$ is the space defined in (4.7) and we use symbol * to denote the solution of the problem in order to avoid confusion between solution and test functions.

We define the errors of the primal and dual problems as follows

$$e_{\tau h} := U^* - U^*_{\tau h} = \{u^* - u^*_{\tau h}, v^* - v^*_{\tau h}\}, \quad \varepsilon_{\tau h} := Z^* - Z^*_{\tau h} = \{w^* - w^*_{\tau h}, z^* - z^*_{\tau h}\}.$$

Since $\mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h}$ is a subspace of $\mathcal{V}_\tau \times \mathcal{V}_\tau$, equation (6.9a) also holds for all functions in $\mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h}$, so we can substitute $Z_{\tau h}$ in (6.9a). Now, substracting it from equation (6.9b), and using the bilinearity of $\mathcal{B}_{CG}(\cdot,\cdot)$ we obtain the Galerkin orthogonality

$$\mathcal{B}_{CG}(e_{\tau h}, Z_{\tau h}) = 0, \ \forall Z_{\tau h} \in \mathcal{V}^{r-1}_{\tau h} \times \mathcal{V}^{r-1}_{\tau h}. \tag{6.11}$$

Now, replacing $U$ by $e_{\tau h} \in \mathscr{U} \times \mathscr{V}$ in (6.10a) and using the Galerkin orthogonality (6.11), we obtain the classical error representation

$$\mathcal{L}(e_{\tau h}) = \mathcal{B}_{CG}(e_{\tau h}, \varepsilon_{\tau h}). \tag{6.12}$$

If we denote by $\mathcal{K} := \Omega_i \times I_j$ each space-time element and $\mathcal{B}_{\mathcal{K}}(\cdot,\cdot)$ the restriction of the bilinear form $\mathcal{B}_{CG}(\cdot,\cdot)$ to each element $\mathcal{K}$, we obtain the following upper bound of the error in terms of local element contributions

$$|\mathcal{L}(e_{\tau h})| = |\mathcal{B}_{CG}(e_{\tau h}, \varepsilon_{\tau h})| = |\sum_{\mathcal{K}} \mathcal{B}_{\mathcal{K}}(e_{\tau h}, \varepsilon_{\tau h})| \leqslant \sum_{\mathcal{K}} |\mathcal{B}_{\mathcal{K}}(e_{\tau h}, \varepsilon_{\tau h})|. \tag{6.13}$$

Equation (6.13) can still be further bounded by a sum over $||e_{\tau h}||_{\mathcal{K}}||\varepsilon_{\tau h}||_{\mathcal{K}}$, where $||\cdot||_{\mathcal{K}}$ is a suitable norm stemming from operator $\mathcal{B}_{\mathcal{K}}$ (in the case where $\mathcal{B}_{\mathcal{K}}$ is positive definite, one would directly select its associated norm). From the mathematical point of view, this sum of norms is the one that should be minimized, but from the engineering point of view, minimizing $\sum_{\mathcal{K}} |\mathcal{B}_{\mathcal{K}}(e_{\tau h}, \varepsilon_{\tau h})|$ often provides better adaptive algorithms (fewer unknowns are needed to achieve a given tolerance error), since this upper bound is sharper than the standard norm-based upper bound [114]. In this work, we display results corresponding to this engineering approach, although we obtain similar results when we consider the upper bound given in terms of $\sum_{\mathcal{K}} ||e_{\tau h}||_{\mathcal{K}}||\varepsilon_{\tau h}||_{\mathcal{K}}$.

## 6.2.2. Alternative error representation

By linearity, we have that $\mathcal{B}_{CG}(U, \varepsilon_{\tau h}) = \mathcal{B}_{CG}(U, Z^*) - \mathcal{B}_{CG}(U, Z^*_{\tau h})$, and from (6.10a) we obtain a weak formulation for the error of the dual problem

$$\begin{cases} \text{Find } \varepsilon_{\tau h} \in \mathcal{V}_\tau \times \mathcal{V}_\tau \text{ such that} \\ \mathcal{B}_{CG}(U, \varepsilon_{\tau h}) = \mathcal{L}(U) - \mathcal{B}_{CG}(U, Z^*_{\tau h}), \ \forall U \in \mathscr{U} \times \mathscr{V}. \end{cases} \tag{6.14}$$

Following [40], we select an alternative bilinear form $\tilde{\mathcal{B}}(\cdot,\cdot)$ in (6.14) in order to obtain a new error $\tilde{\varepsilon}_{\tau h}$ as the solution of the following problem

$$\begin{cases} \text{Find } \tilde{\varepsilon}_{\tau h} \in \mathcal{V}_\tau \times \mathcal{V}_\tau \text{ such that} \\ \tilde{\mathcal{B}}(U, \tilde{\varepsilon}_{\tau h}) = \mathcal{L}(U) - \mathcal{B}_{CG}(U, Z^*_{\tau h}),\ \forall U \in \mathscr{U} \times \mathscr{V}. \end{cases} \tag{6.15}$$

From (6.10b), we see that this alternative bilinear form also satisfies the Galerkin orthogonality

$$\tilde{\mathcal{B}}(U_{\tau h}, \tilde{\varepsilon}_{\tau h}) = \mathcal{L}(U_{\tau h}) - \mathcal{B}_{CG}(U_{\tau h}, Z^*_{\tau h}) = 0,\ \forall U_{\tau h} \in \mathfrak{U}_{\tau h} \times \mathfrak{U}_{\tau h}.$$

Now, replacing $U$ by $e_{\tau h}$ in (6.15) and using the Galerkin orthogonality (6.11), we obtain the new error representation

$$\mathcal{L}(e_{\tau h}) = \tilde{\mathcal{B}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h}), \tag{6.16}$$

and, as in (6.13), we have the upper bound

$$|\mathcal{L}(e_{\tau h})| = |\tilde{\mathcal{B}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})| = |\sum_{\mathcal{K}} \tilde{\mathcal{B}}_{\mathcal{K}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})| \leqslant \sum_{\mathcal{K}} |\tilde{\mathcal{B}}_{\mathcal{K}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|. \tag{6.17}$$

In [40], the authors show numerically that for the Helmholtz equation in 1D, there exist alternative bilinear forms for which the upper bounds of the new error representation are sharper than the classical ones. This fact leads us to think that if we find a suitable alternative bilinear form $\tilde{\mathcal{B}}(\cdot,\cdot)$, the upper bound (6.17) could also be sharper than (6.13) for time-domain problems. The resulting method is a better guidance criterion for the adaptive process.

Following [40], we select the alternative bilinear form $\tilde{\mathcal{B}}(\cdot,\cdot)$ by modifying from $\mathcal{B}_{CG}(\cdot,\cdot)$ the terms coming from the second order time derivative

$$\begin{aligned} \tilde{\mathcal{B}}(U, Z) = \beta \int_I (u_t, w)dt - \int_I (v, w)dt + \beta(u(0), w(0)) \\ + \beta \int_I \langle v_t, z \rangle\, dt + \int_I (\alpha u_x, z_x)dt + \beta(v(0), z(0)), \end{aligned} \tag{6.18}$$

where $\beta \in \mathbb{R} - \{0\}$.

Following an analogous process to that described in Section 6.2, we conclude that the strong formulation of the unconventional dual problem

$$\begin{cases} \text{Find } \tilde{Z} = \{\tilde{w}, \tilde{z}\} \in \mathcal{V}_\tau \times \mathcal{V}_\tau \text{ such that} \\ \tilde{\mathcal{B}}(U, \tilde{Z}) = \mathcal{L}(U),\ \forall U = \{u, v\} \in \mathscr{U} \times \mathscr{V}. \end{cases} \tag{6.19}$$

is given by

$$\begin{cases} \beta\tilde{z}_t = -\tilde{w} & \text{in } \Omega \times I, \\ -\beta\tilde{w}_t - (\alpha\tilde{z}_x)_x = g & \text{in } \Omega \times I, \\ \tilde{z}(a,t) = \tilde{z}(b,t) = 0 & \text{in } I, \\ \beta\tilde{z}(x,T) = \tilde{z}_T(x) & \text{in } \Omega, \\ \beta\tilde{w}(x,T) = \tilde{w}_T(x) & \text{in } \Omega, \end{cases} \tag{6.20}$$

We know that the solution of the homogeneous wave equation $u_{tt} - \alpha u_{xx} = 0$ is of the form

$$u(x,t) = \sum_{n=0}^{\infty} \left( c_n \cos\left(\frac{n\pi\sqrt{\alpha}t}{l}\right) + d_n \sin\left(\frac{n\pi\sqrt{\alpha}t}{l}\right)\right) \sqrt{\frac{2}{l}} \sin\left(\frac{n\pi x}{l}\right)$$

where $l$ is the length of the spatial interval $\Omega = (a,b)$ and $c_n$ and $d_n$ are constants that depend on the initial conditions of the equation.

Problem (6.20) verifies equation $\tilde{z}_{tt} - \dfrac{1}{\beta^2}(\alpha\tilde{z}_x)_x = \dfrac{1}{\beta^2}\ g$, so when $\beta$ is large enough, the solution of (6.20) converges to the solution of its associated homogeneous equation. Moreover, when $\beta$ is large enough, $\tilde{z}_{tt}$ is close to zero, and therefore, the solution $z(x,t)$ is linear in time. In conclusion, with the alternative bilinear form (6.18), we are selecting unconventional dual problems that are almost linear in time.

## 6.3. Tensor product space-time refinements

We work with two meshes: a coarse mesh and a reference mesh. We perform optimal refinements over the coarse mesh, while the reference mesh is employed to calculate the reference solutions of problems (6.9a) and (6.10a).

### 6.3.1. Uniform $\tau h-$refinements

In Algorithm 2, we perform uniform refinements in space ($h-$refinements) and time ($\tau-$refinements) simultaneously. The inputs of the algorithm are the endpoints $a$ and $b$ of $\Omega$, the final instant $T$, the number of global refinements $n_{refin}$ and the number of elements in space $n_{coarse}$ and time $m_{coarse}$ of the coarse mesh. First, we calculate the primal (6.9a) and the dual (6.10a) reference solutions. We solve the primal problem (6.9b) and dual problem (6.10b) in the coarse mesh and also in a sequence of uniformly refined meshes. Then, we inject the primal and dual solutions over the reference mesh. According to the discrete spaces selected in Section 6.1, the injection of the primal problem is performed by piecewise linear interpolation both in space and time, whereas the injection of the dual problem

is performed by piecewise linear interpolation in space and piecewise constant interpolation in time. Finally, we estimate the errors $e_{\tau h}$ and $\varepsilon_{\tau h}$ and solve the unconventional dual problem (6.15) to estimate $\tilde{\varepsilon}_{\tau h}$.

**Algorithm 2** Uniform $\tau h-$refinements

**Input:** $a$, $b$, $T$, $n_{refin}$, $n_{coarse}$, $m_{coarse}$
$h_{ref} \leftarrow \frac{b-a}{n_{coarse}\cdot 2^{n_{refin}}}$
$\tau_{ref} \leftarrow \frac{T}{m_{coarse}\cdot 2^{n_{refin}}}$
$U_{\tau h}^{ref} \leftarrow$ Primal($h_{ref}$, $\tau_{ref}$) ▷ Compute the reference solutions
$Z_{\tau h}^{ref} \leftarrow$ Dual($h_{ref}$, $\tau_{ref}$)
**for** $i = 0$ to $n_{refin} - 1$ **do**
    $h \leftarrow \frac{b-a}{n_{coarse}\cdot 2^{i}}$
    $\tau \leftarrow \frac{T}{m_{coarse}\cdot 2^{i}}$
    $U_{\tau h} \leftarrow$ Primal($h$, $\tau$)
    $Z_{\tau h} \leftarrow$ Dual($h$, $\tau$)
    $U_{\tau h}^{inj} \leftarrow$ Inject($h_{ref}$, $\tau_{ref}$, $U_{\tau h}$) ▷ Injections over the reference mesh
    $Z_{\tau h}^{inj} \leftarrow$ Inject($h_{ref}$, $\tau_{ref}$, $Z_{\tau h}$)
    $\tilde{\varepsilon}_{\tau h} \leftarrow$ NewDual($h_{ref}$, $\tau_{ref}$, $Z_{\tau h}^{inj}$) ▷ Estimate the errors
    $e_{\tau h} \leftarrow U_{\tau h}^{ref} - U_{\tau h}^{inj}$
    $\varepsilon_{\tau h} \leftarrow Z_{\tau h}^{ref} - Z_{\tau h}^{inj}$
**end for**

### 6.3.2. Goal-oriented $\tau-$ and $h-$adaptivity

Algorithm 3 performs local refinements in space for all $t \in I$ and, simultaneously, local refinements in time for all $x \in \Omega$. In other words, we restrict to tensor-product type refinements. We use the following upper bound of the error in terms of local contributions in time and space separately

$$|\mathcal{L}(e_{\tau h})| = |\mathcal{B}_{CG}(e_{\tau h}, \varepsilon_{\tau h})| \leqslant \frac{1}{2}\sum_{i=1}^{n} |\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})| + \frac{1}{2}\sum_{j=1}^{m} |\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|, \quad (6.21)$$

where $\mathcal{B}_{\Omega_i}(\cdot,\cdot)$ and $\mathcal{B}_{I_j}(\cdot,\cdot)$ are restrictions of the bilinear form $\mathcal{B}_{CG}(\cdot,\cdot)$ to each element $\Omega_i \times I$ and $\Omega \times I_j$, respectively.

The inputs of the algorithm are the endpoints $a$ and $b$, the final instant $T$, the tolerances $tol_1$ and $tol_2$, and the number of elements in space $n$, $n_{ref}$ and time $m$, $m_{ref}$ of the coarse and reference meshes, respectively.

First, we calculate the primal (6.9a) and dual (6.10a) reference solutions. Then, we solve the primal (6.9b) and dual (6.10b) problems in the coarse mesh and we

estimate the relative error in the QoI, which is used as a stopping criterion. For each element, we estimate the following quantities according to the upper bound (6.21)

$$\frac{|\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|}{\max\left(\max_{\Omega_i}|\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|, \max_{I_j}|\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|\right)} \cdot 100, \tag{6.22}$$

$$\frac{|\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|}{\max\left(\max_{\Omega_i}|\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|, \max_{I_j}|\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|\right)} \cdot 100. \tag{6.23}$$

We refine those elements in which the above quantities are greater than or equal to $tol_2$. Finally, the obtained adapted mesh becomes the coarse grid for the next iteration. The adaptive process ends when the relative error of $\mathcal{L}(U)$ is below $tol_1$ or when $h$ or $\tau$ of the adapted mesh is smaller than $h_{ref}$ or $\tau_{ref}$ of the reference mesh, that is, when the refinement process reaches the resolution of the reference solution in either space or time.

Using (6.21), we consider a new upper bound

$$|\mathcal{L}(e_{\tau h})| = |\tilde{\mathcal{B}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})| \leqslant \frac{1}{2}\sum_{i=1}^{n}|\tilde{\mathcal{B}}_{\Omega_i}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})| + \frac{1}{2}\sum_{j=1}^{m}|\tilde{\mathcal{B}}_{I_j}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|. \tag{6.24}$$

Then, we employ the following quantities as the criteria to guide the adaptive process in Algorithm 3

$$\frac{|\tilde{\mathcal{B}}_{\Omega_i}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|}{\max\left(\max_{\Omega_i}|\tilde{\mathcal{B}}_{\Omega_i}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|, \max_{I_j}|\tilde{\mathcal{B}}_{I_j}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|\right)} \cdot 100, \tag{6.25}$$

$$\frac{|\tilde{\mathcal{B}}_{I_j}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|}{\max\left(\max_{\Omega_i}|\tilde{\mathcal{B}}_{\Omega_i}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|, \max_{I_j}|\tilde{\mathcal{B}}_{I_j}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})|\right)} \cdot 100. \tag{6.26}$$

**Algorithm 3** Goal-oriented $\tau-$ and $h-$adaptivity

**Input:** $a$, $b$, $T$, $tol_1$, $tol_2$, $n$, $m$, $n_{ref}$, $m_{ref}$
$h_{ref} \leftarrow \frac{b-a}{n_{ref}}$, $\tau_{ref} \leftarrow \frac{T}{m_{ref}}$
$U_{\tau h}^{ref} \leftarrow$ Primal($h_{ref}$, $\tau_{ref}$) ▷ Compute the reference solutions
$Z_{\tau h}^{ref} \leftarrow$ Dual($h_{ref}$, $\tau_{ref}$)
$h \leftarrow \frac{b-a}{n}$, $\tau \leftarrow \frac{T}{m}$ ▷ Initialization
$rel\_error \leftarrow 1$
**while** $rel\_error \geqslant tol_1$ & $min(h) > h_{ref}$ & $min(\tau) > \tau_{ref}$ **do**
    $U_{\tau h} \leftarrow$ Primal($h$, $\tau$) ▷ Compute the coarse solutions
    $Z_{\tau h} \leftarrow$ Dual($h$, $\tau$)
    $U_{\tau h}^{inj} \leftarrow$ Inject($h_{ref}$, $\tau_{ref}$, $U_{\tau h}$) ▷ Injections over the reference mesh
    $Z_{\tau h}^{inj} \leftarrow$ Inject($h_{ref}$, $\tau_{ref}$, $Z_{\tau h}$)
    $e_{\tau h} \leftarrow U_{\tau h}^{ref} - U_{\tau h}^{inj}$ ▷ Estimate the errors
    $\varepsilon_{\tau h} \leftarrow Z_{\tau h}^{ref} - Z_{\tau h}^{inj}$
    $\tilde{\varepsilon}_{\tau h} \leftarrow$ NewDual($h_{ref}$, $\tau_{ref}$, $Z_{\tau h}^{inj}$)
    $rel\_error \leftarrow \frac{|\mathcal{L}(e_{\tau h})|}{|\mathcal{L}(U_{\tau h}^{ref})|} \cdot 100$
    **if** $rel\_error \geqslant tol_1$ **then**
        **for** $i = 1$ to $n$ **do** ▷ Local refinements in space
            $test\_space = \frac{|\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|}{\max\left(\max\limits_{\Omega_i} |\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|, \max\limits_{I_j} |\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|\right)} \cdot 100$
            **if** $test\_space \geqslant tol_2$ **then**
                $refine\ \ \Omega_i$
            **end if**
        **end for**
        **for** $j = 1$ to $m$ **do** ▷ Local refinements in time
            $test\_time = \frac{|\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|}{\max\left(\max\limits_{\Omega_i} |\mathcal{B}_{\Omega_i}(e_{\tau h}, \varepsilon_{\tau h})|, \max\limits_{I_j} |\mathcal{B}_{I_j}(e_{\tau h}, \varepsilon_{\tau h})|\right)} \cdot 100$
            **if** $test\_time \geqslant tol_2$ **then**
                $refine\ \ I_j$
            **end if**
        **end for**
    **end if**
**end while**

## 6.4. Numerical results

### 6.4.1. Uniform space-time refinements

Let $\Omega = (-1, 1)$ and $I = (0, 2]$, we consider problem (6.1) with $f(x,t) = \alpha\cos(3\pi t)$, $u_0(x) = \phi(x)$ and $v_0(x) = 0$ where $\sqrt{\alpha} = \frac{3\pi}{5}$ and

$$\phi(x) = \frac{\cos(5x)\sec(5) - 1}{25}.$$

We select the following output functional

$$\mathcal{L}_0(u) = \int_I \int_\Omega u(x,t)\alpha\cos(3\pi t)dxdt, \quad \mathcal{L}_1(v) = \int_\Omega v(x,T)\phi(x)dx.$$

With this data, the source term of the dual problem is the same as in the primal problem and $\varphi(x,T) = u(x,0) = \phi(x)$ (see Section 6.2), and therefore, the solutions of the primal and dual problems are such that $U(x,t) = \Psi(x, T-t)$, i.e, are the same function but reversed in time.

We set a coarse mesh with $2^3$ elements in space and time, respectively. Then, we perform 4 global uniform refinements so the reference mesh has $2^7$ elements in each variable ($2^{14}$ elements in total). Figure 6.1 presents the primal (forward in time) solution, which coincides with the dual one that moves backwards in time.

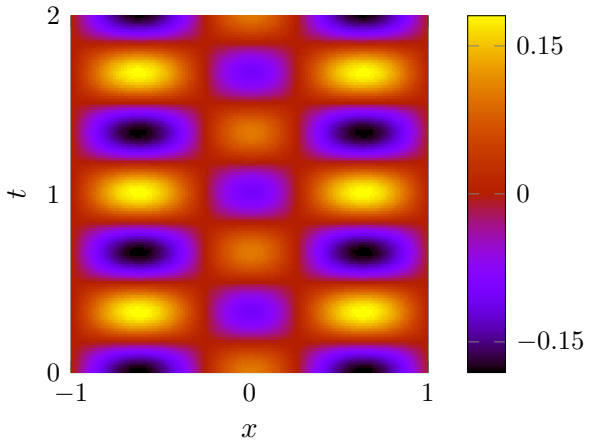



Figure 6.1.: Solution of the primal (forwards in time) and dual (backwards in time) problems.

Figure 6.2 shows the upper bounds obtained with a mesh of $2^6$ elements (the coarse one) when we vary the values of the parameter $\beta \in \mathbb{R} - \{0\}$ of $\tilde{\mathcal{B}}(\cdot,\cdot)$ described in (6.18).

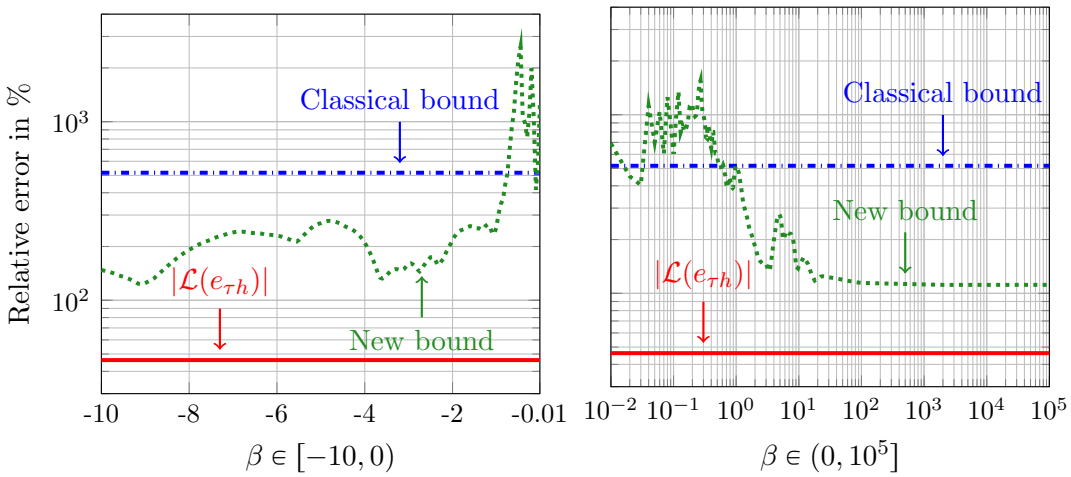



Figure 6.2.: Upper bounds obtained in a given mesh varying the parameter $\beta$ in (6.18) considering negative (left) and positive (right) values.

We can see that when $\beta = 1$ the bounds coincide, and when $\beta \in (0, 1)$ the classical upper bound is, in general, sharper than the new one. This happens because the propagation speed of the wave and the source term of problem (6.20) are much higher so the numerical problem becomes unstable. However, when $\beta > 1$, the pseudo-dual problem exhibits better stability properties than the classical dual problem as its speed of propagation is smaller and, for large values of $\beta$, the stability constant approaches one at the discrete level. We can see that when $\beta > 1$, the new upper bound is sharper than the classical one, i.e.,

$$|\mathcal{L}(e_{\tau h})| \leqslant \sum_{\mathcal{K}} |\tilde{\mathcal{B}}_{\mathcal{K}}(e_{\tau h}, \tilde{\varepsilon}_{\tau h})| \leqslant \sum_{\mathcal{K}} |\mathcal{B}_{\mathcal{K}}(e_{\tau h}, \varepsilon_{\tau h})|.$$

Figure 6.2 shows a similar pattern for negative values of parameter $\beta$.

Figure 6.3 shows the solution of the unconventional dual problem (6.19) and its derivative with $\beta = 10^2$. We observe that the solution is constant in time and its derivative is close to zero.

Figures 6.4, 6.5 and 6.6 show the errors of the primal, dual and unconventional dual problems, respectively. Although the primal and dual solutions are the same, errors $e_{\tau h}$ and $\varepsilon_{\tau h}$ are different because they belong to different spaces. When $\beta = 1$, errors $\varepsilon_{\tau h}$ and $\tilde{\varepsilon}_{\tau h}$ coincide because $\tilde{\mathcal{B}}(\cdot,\cdot) = \mathcal{B}_{CG}(\cdot,\cdot)$ (see Section 6.2).

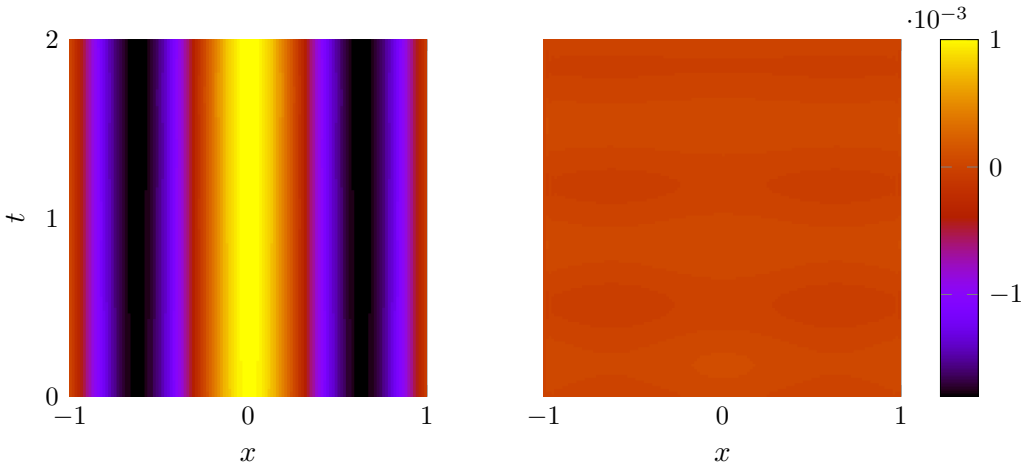
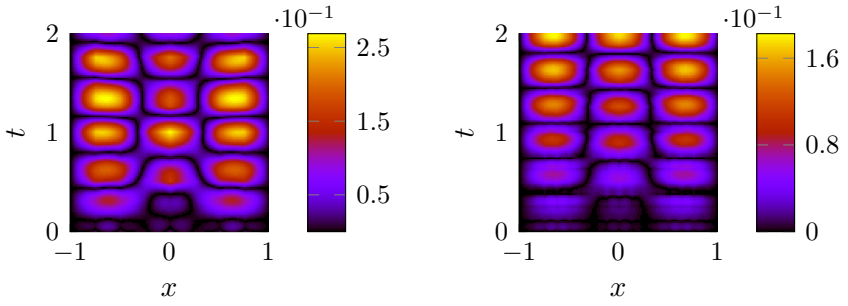


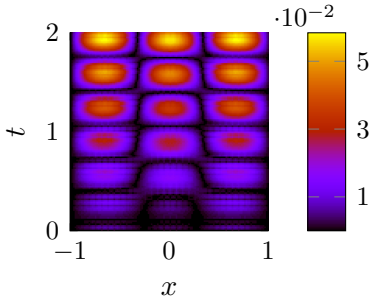

Figure 6.3.: Solution of the pseudo-dual problem (6.19) (left) and its time derivative (right) with $\beta = 10^2$.



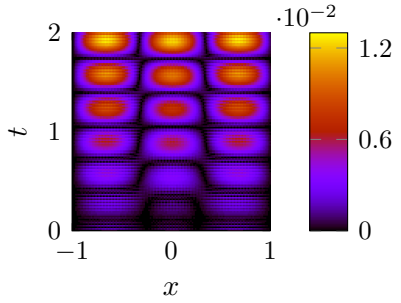

(a) Coarse mesh $|e_{\tau h}|$ error.




(b) Error $|e_{\tau h}|$ on mesh 1.

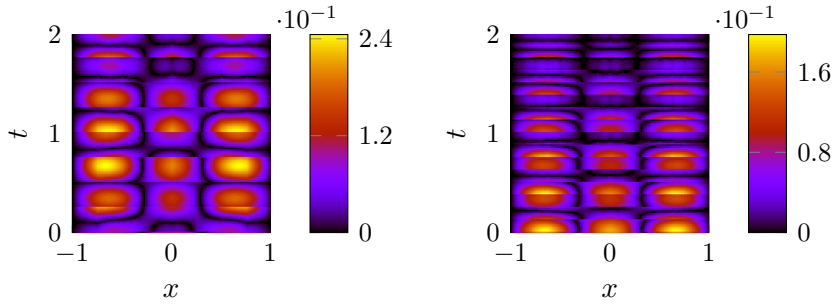



(c) Error $|e_{\tau h}|$ on mesh 2.




(d) Error $|e_{\tau h}|$ on mesh 3.

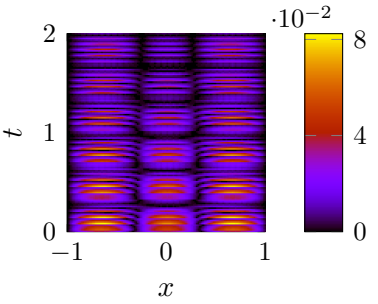
Figure 6.4.: Error of the primal problem with $\beta = 10^2$ in different grids.



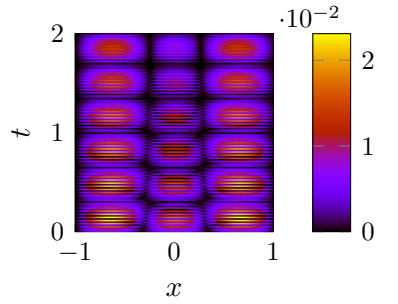

(a) Coarse mesh $|\varepsilon_{\tau h}|$ error.




(b) Error $|\varepsilon_{\tau h}|$ on mesh 1.

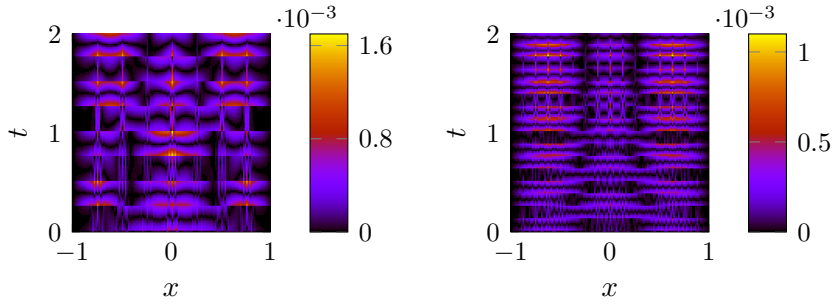



(c) Error $|\varepsilon_{\tau h}|$ on mesh 2.




(d) Error $|\varepsilon_{\tau h}|$ on mesh 3.

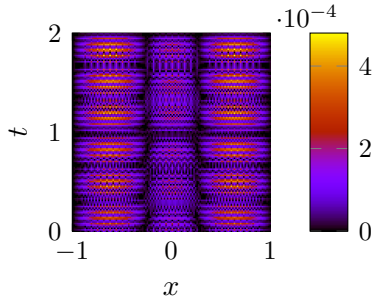
Figure 6.5.: Error of the dual problem with $\beta = 10^2$ in different grids.



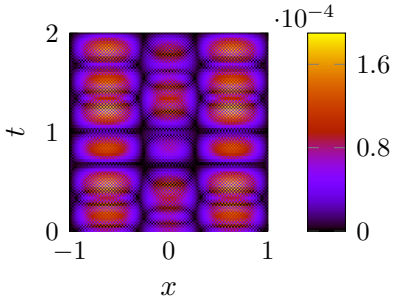

(a) Coarse mesh $|\tilde{\varepsilon}_{\tau h}|$ error.




(b) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 1.




(c) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 2.




(d) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 3.

Figure 6.6.: Solution of problem (6.15) with $\beta = 10^2$ in different grids.

Finally, Figure 6.7 compares the upper bounds (6.13) and (6.17) of $|\mathcal{L}(e_{\tau h})|$ and we observe that for this problem, the new upper bound is sharper than the classical one.

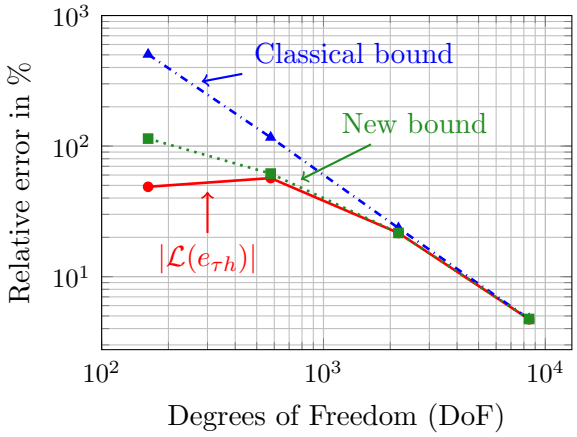



Figure 6.7.: Upper bounds (6.13) and (6.17) with $\beta = 10^2$.

Now, we consider the following output functional

$$\mathcal{L}_0(u) = \int_I \int_{\Omega_0} u(x,t)\cos(3\pi t)dxdt, \quad \mathcal{L}_1(v) = 0,$$

where $\Omega_0 = [-0.5, -0.25]$. In this case, the QoI is a weighted solution over a small subdomain of $\Omega$.

Figure 6.8 displays the primal and dual reference solutions. Figures 6.9 and 6.10 show the errors of the dual and unconventional dual problems, respectively.

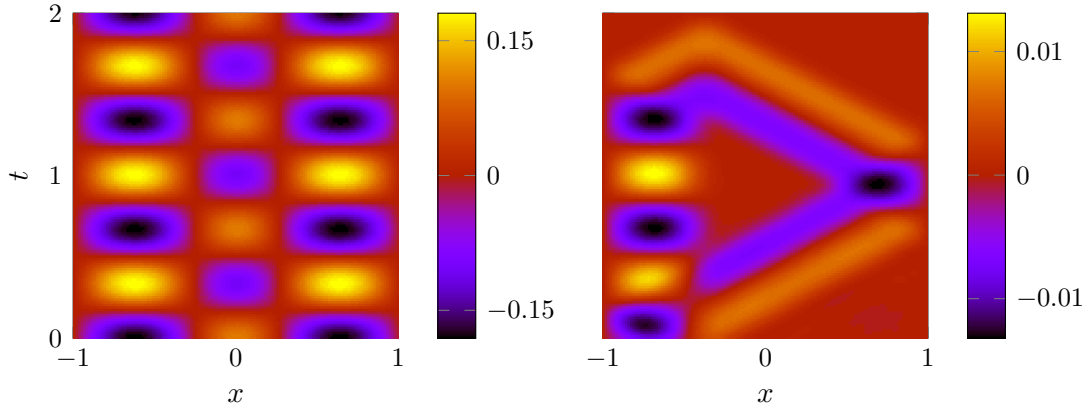
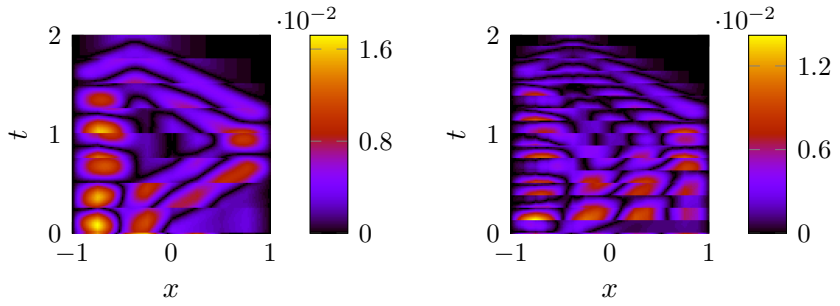


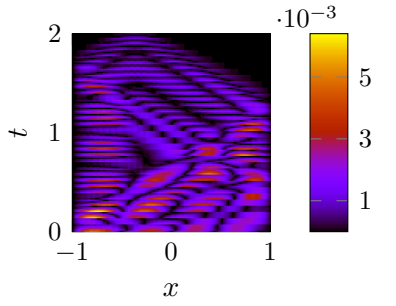

Figure 6.8.: Solution of the primal (left) and dual (right) problems.



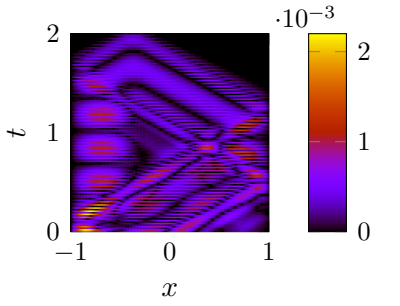

(a) Coarse mesh $|\varepsilon_{\tau h}|$ error.




(b) Error $|\varepsilon_{\tau h}|$ on mesh 1.

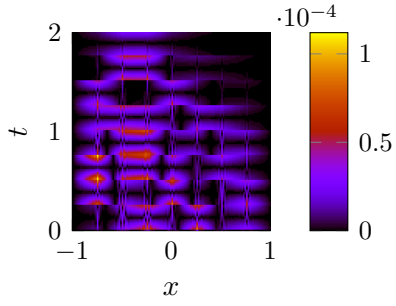



(c) Error $|\varepsilon_{\tau h}|$ on mesh 2.

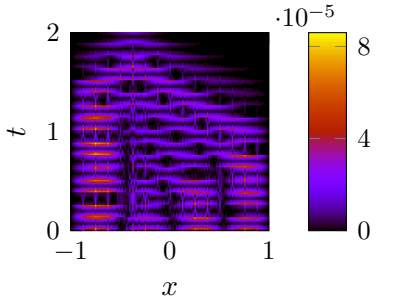



(d) Error $|\varepsilon_{\tau h}|$ on mesh 3.

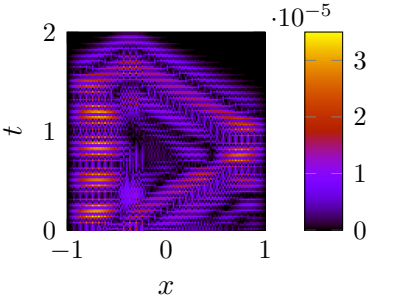
Figure 6.9.: Error of the dual problem with $\beta = 10^2$ in each refinement.



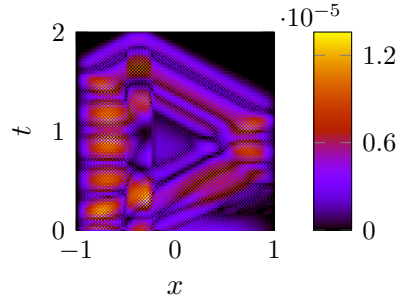

(a) Coarse mesh $|\tilde{\varepsilon}_{\tau h}|$ error.




(b) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 1.




(c) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 2.




(d) Error $|\tilde{\varepsilon}_{\tau h}|$ on mesh 3.

Figure 6.10.: Solution of problem (6.15) with $\beta = 10^2$ in each refinement.

Finally, Figure 6.11 exhibits the upper bounds. We conclude that for this case, the new upper bound is also sharper than the classical one.

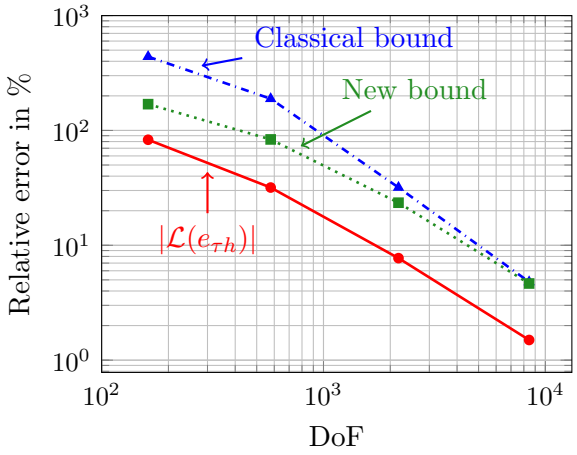



Figure 6.11.: Upper bounds (6.13) and (6.17) with $\beta = 10^2$.

## 6.4.2. Goal-oriented space-time refinements

Let $\Omega = (-2, 2)$ and $I = (0, 6]$, we select $u_0(x) = v_0(x) = 0$ and we define the source term and the wave speed piecewise as

$$f(x,t) = \begin{cases} 2\cos(2\pi t), & x \in [0.5, 1.5]\,, \quad t \in [0, 1.5]\,, \\ 0, & \text{elsewhere.} \end{cases}$$

$$\alpha(x) = \begin{cases} \dfrac{1}{16}, & x \leqslant 0, \\ 1, & x > 0. \end{cases}$$

We consider the output functional

$$\mathcal{L}_0(u) = \int_I \int_{\Omega_0} \cos(2\pi t) u(x,t) dx dt, \quad \mathcal{L}_1(v) = 0,$$

where $\Omega_0 = [0.5, 1.5]$. We select $\beta = 10^2$ and perform the mesh adaptivity by executing Algorithm 3. We set a coarse mesh with $2^6$ elements and a reference mesh with $2^{14}$ elements. We also set the tolerances: $tol_1 = 0.5\%$ and $tol_2 = 5\%$.

Figure 6.12 displays the primal and dual reference solutions. Figure 6.13 shows the final adapted meshes when we use classical and alternative criteria to perform adaptivity.

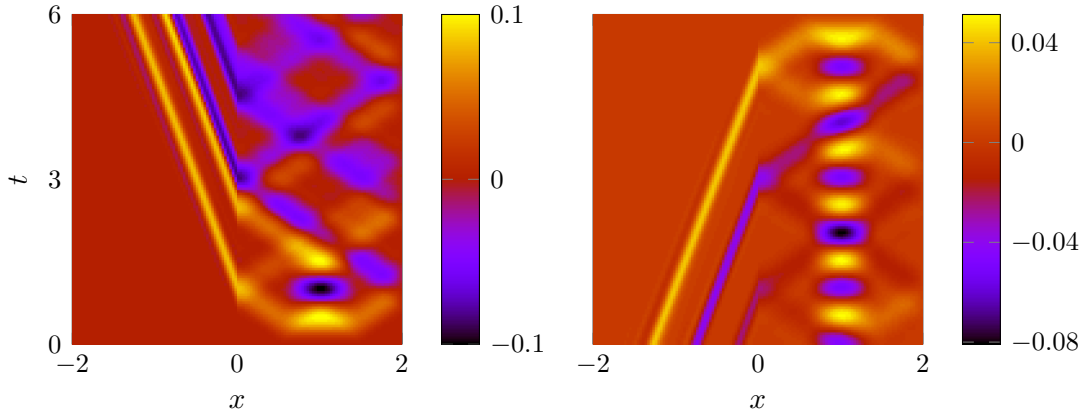



Figure 6.12.: Solution of the primal (left) and dual (right) problems.

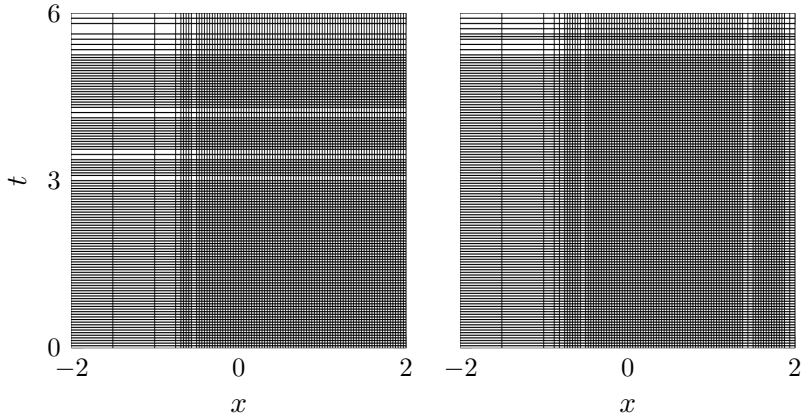



Figure 6.13.: Adapted space-time meshes using classical criterion ((6.22),(6.23)) (left) and alternative criterion ((6.25),(6.26)) (right).

Figures 6.14 and 6.16 show the upper bounds (6.21) and (6.24) when we use the classical criteria (6.22) and (6.23), and the alternative criteria (6.25) and (6.26), respectively. In both cases, the new upper bound is sharper than the classical one. Finally, Figures 6.15 and 6.17 display the spatial and temporal contributions of the upper bounds (6.21) and (6.24). We conclude that, in this case, the temporal contribution is almost independent of the use of bound (6.21) or (6.24), but the spatial contribution is sharper when employing the new bound.

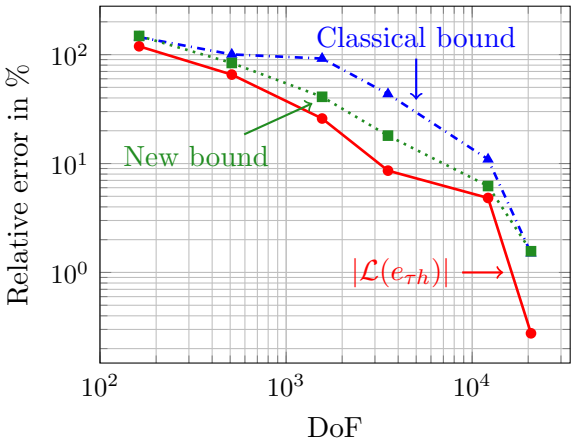



Figure 6.14.: Upper bounds (6.21) and (6.24) using classical criteria (6.22) and (6.23) in Algorithm 3.

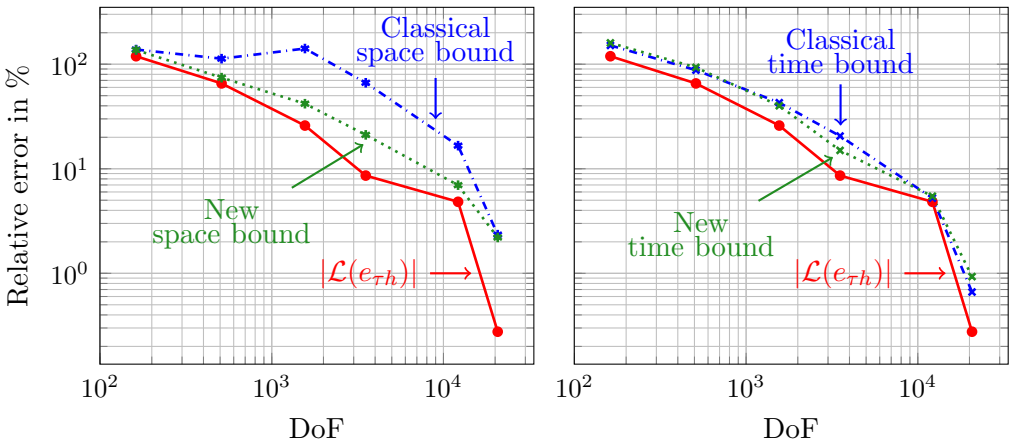



Figure 6.15.: Spatial contribution (left) and temporal contribution (right) of the upper bounds (6.21) and (6.24) using classical criteria (6.22) and (6.23) in Algorithm 3.

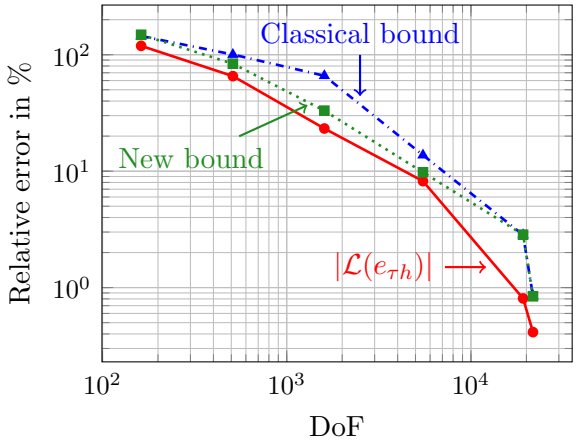



Figure 6.16.: Upper bounds (6.21) and (6.24) using alternative criteria (6.25) and (6.26) in Algorithm 3.

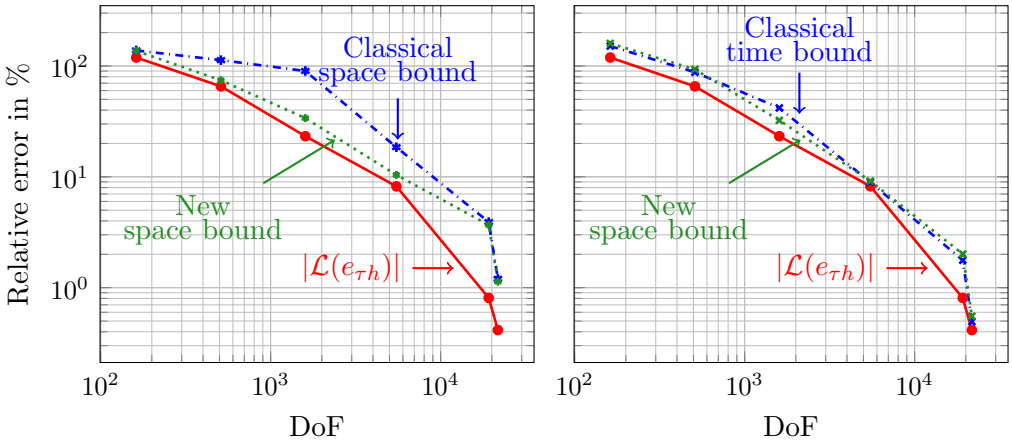



Figure 6.17.: Spatial contribution (left) and temporal contribution (right) of the upper bounds (6.21) and (6.24) using alternative criteria (6.25) and (6.26) in Algorithm 3.

In conclusion, we propose an unconventional error representation for GOA in time-domain. We compare the classical upper bounds estimates of the error in the quantity of interest with the new ones, observing that the new bounds are sharper when applied to the 1D wave equation.

# 7. Isogeometric Residual Minimization (iGRM) method with direction splitting

When problems are unstable, adaptive algorithms may and in general will introduce refinements in unnecessary places. To overcome these situations, it would be desirable to employ stable methods. In this chapter, we combine three modern methods, namely: Isogeometric Analysis (IGA), residual minimization methods and Alternating Direction Implicit (ADI) schemes in order to achieve accuracy, stability and efficiency, respectively. We call our method isogeometric Residual Minimization (iGRM) with direction splitting. We explore how to obtain suitable stable formulations that can accommodate goal-oriented adaptive algorithms as the ones proposed in previous chapters.

First, we describe three second-order ADI schemes for the 2D and 3D+time advection diffusion-equation together with tensor product B-spline basis functions in space. Then, we apply a residual minimization method at each time step in order to stabilize the method. Finally, we show that the resulting system could be factorized as tensor product of $1D$ matrices in space and therefore we can solve it with linear computational cost of the direct solver. We check the convergence order with a manufactured solution problem and we illustrate the performance of our method with a circular wind problem in 2D+time and a pollution propagation problem in 3D+time.

## 7.1. Model problem and ADI methods

We consider problem (2.1) over a rectangular domain in space, i.e., $\Omega = \Omega_x \times \Omega_y \subset \mathbb{R}^2$ or $\Omega = \Omega_x \times \Omega_y \times \Omega_z \subset \mathbb{R}^3$, where $\Omega_x$, $\Omega_y$ and $\Omega_z$ are intervals in $\mathbb{R}$. We also consider constant diffusivity $\nu$ and a velocity field of the form $\beta = [\beta_x \;\; \beta_y]$ in $2D$ and $\beta = [\beta_x \;\; \beta_y \;\; \beta_z]$ in $3D$.

Now, we split the advection-diffusion operator $\mathcal{L}u := -\nabla \cdot (\nu \nabla u) + \beta \cdot \nabla u$ as $\mathcal{L}u = \mathcal{L}_1 u + \mathcal{L}_2 u$ in $2D$ and $\mathcal{L}u = \mathcal{L}_1 u + \mathcal{L}_2 u + \mathcal{L}_3 u$ in $3D$ where

$$\mathcal{L}_1 u := -\nu \frac{\partial u}{\partial x^2} + \beta_x \frac{\partial u}{\partial x}, \;\; \mathcal{L}_2 u := -\nu \frac{\partial u}{\partial y^2} + \beta_y \frac{\partial u}{\partial y}, \;\; \mathcal{L}_3 u := -\nu \frac{\partial u}{\partial z^2} + \beta_z \frac{\partial u}{\partial z}. \quad (7.1)$$

We select the time partition defined in (2.4) with uniform time-step size $\tau$. Based on operator splitting (7.1), we consider different ADI schemes to discretize in time problem (2.1). We denote by $u^k$ the approximated solution of $u$ at each time step $t_k, \forall k = 1, \ldots, m$. For the space discretization, we approximate the solution of a fixed time step with tensor product of one-dimensional B-spline basis functions [39, 81] of order $p \geqslant 1$ in 2D and 3D as

$$u_h^k(x, y) = \sum_{i,j} u_{ij}^k B_{i;p}(x) B_{j;p}(y), \tag{7.2a}$$

$$u_h^k(x, y, z) = \sum_{i,j,l} u_{ijl}^k B_{i;p}(x) B_{j;p}(y) B_{l;p}(z). \tag{7.2b}$$

### 7.1.1. Strang splitting scheme with Crank-Nicolson

We consider the Strang splitting scheme [52, 132] for the 2D+time advection-diffusion equation. First, we divide problem $u_t + \mathcal{L}u = f$ into two subproblems

$$\begin{cases} P_1 : u_t + \mathcal{L}_1 u = f, \\ P_2 : u_t + \mathcal{L}_2 u = 0, \end{cases}$$

and the scheme integrates the solution from $t_k$ to $t_{k+1}$, $\forall k = 1, \ldots, m$ in three substeps

$$\begin{cases} \text{Solve } P_1 : u_t + \mathcal{L}_1 u = f, \text{ in } (t_k, t_{k+1/2}), \\ \text{Solve } P_2 : u_t + \mathcal{L}_2 u = 0, \text{ in } (t_k, t_{k+1}), \\ \text{Solve } P_1 : u_t + \mathcal{L}_1 u = f, \text{ in } (t_{k+1/2}, t_{k+1}). \end{cases} \tag{7.3}$$

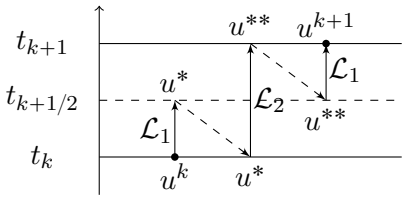



Figure 7.1.: Interpretation of the Strang splitting scheme.

We can employ different methods in each substep of (7.3). In particular, if we select the Crank-Nicolson method in (7.3), we obtain

$$\begin{cases} \dfrac{u^* - u^k}{\tau/2} + \dfrac{1}{2}\left(\mathcal{L}_1 u^* + \mathcal{L}_1 u^k\right) = \dfrac{1}{2}\left(f^{k+1/2} + f^k\right), \\ \dfrac{u^{**} - u^*}{\tau} + \dfrac{1}{2}\left(\mathcal{L}_2 u^{**} + \mathcal{L}_2 u^*\right) = 0, \\ \dfrac{u^{k+1} - u^{**}}{\tau/2} + \dfrac{1}{2}\left(\mathcal{L}_1 u^{k+1} + \mathcal{L}_1 u^{**}\right) = \dfrac{1}{2}\left(f^{k+1} + f^{k+1/2}\right), \end{cases} \tag{7.4}$$

where $u^*$ and $u^{**}$ denote the intermediate solutions defined in the Strang integration scheme (see the interpretation in Figure 7.1).

We derive a variational formulation of scheme (7.4) as

$$\begin{cases} (u^*, v) + \dfrac{\tau}{4}\left(\nu\dfrac{\partial u^*}{\partial x}, \dfrac{\partial v}{\partial x}\right) + \dfrac{\tau}{4}\left(\beta_x\dfrac{\partial u^*}{\partial x}, v\right) \\ \qquad = \left(u^k, v\right) - \dfrac{\tau}{4}\left(\nu\dfrac{\partial u^k}{\partial x}, \dfrac{\partial v}{\partial x}\right) - \dfrac{\tau}{4}\left(\beta_x\dfrac{\partial u^k}{\partial x}, v\right) \\ \qquad + \dfrac{\tau}{4}\left(f^{k+1/2} + f^k, v\right), \\ \\ (u^{**}, v) + \dfrac{\tau}{2}\left(\nu\dfrac{\partial u^{**}}{\partial y}, \dfrac{\partial v}{\partial y}\right) + \dfrac{\tau}{2}\left(\beta_y\dfrac{\partial u^{**}}{\partial y}, v\right) \\ \qquad = (u^*, v) - \dfrac{\tau}{2}\left(\nu\dfrac{\partial u^*}{\partial y}, \dfrac{\partial v}{\partial y}\right) - \dfrac{\tau}{2}\left(\beta_y\dfrac{\partial u^*}{\partial y}, v\right), \\ \\ \left(u^{k+1}, v\right) + \dfrac{\tau}{4}\left(\nu\dfrac{\partial u^{k+1}}{\partial x}, \dfrac{\partial v}{\partial x}\right) + \dfrac{\tau}{4}\left(\beta_x\dfrac{\partial u^{k+1}}{\partial x}, v\right) \\ \qquad = (u^{**}, v) - \dfrac{\tau}{4}\left(\nu\dfrac{\partial u^{**}}{\partial x}, \dfrac{\partial v}{\partial x}\right) - \dfrac{\tau}{4}\left(\beta_x\dfrac{\partial u^{**}}{\partial x}, v\right) \\ \qquad + \dfrac{\tau}{4}\left(f^{k+1} + f^{k+1/2}, v\right), \end{cases} \tag{7.5}$$

where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product.

Finally, employing approximation (7.2a) for both trial and test functions, we express problem (7.5) in matrix form as

$$\begin{cases} \left[M_p^x + \dfrac{\tau}{4}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u^* \\ \qquad = \left[M_p^x - \dfrac{\tau}{4}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u^k + \dfrac{\tau}{4}\left(F^{k+1/2} + F^k\right), \\ M_p^x \otimes \left[M_p^y + \dfrac{\tau}{2}\left(K_p^y + R_p^y\right)\right] u^{**} = M_p^x \otimes \left[M_p^y - \dfrac{\tau}{2}\left(K_p^y + R_p^y\right)\right] u^*, \\ \left[M_p^x + \dfrac{\tau}{4}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u^{k+1} \\ \qquad = \left[M_p^x - \dfrac{\tau}{4}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u^{**} + \dfrac{\tau}{4}\left(F^{k+1} + F^{k+1/2}\right), \end{cases} \tag{7.6}$$

where $M_p^{x,y}$, $K_p^{x,y}$ and $R_p^{x,y}$ are the 1D mass, stiffness and advection matrices of order $p$, respectively. Here, $\otimes$ denotes the Kronecker product of matrices.

We will see in the numerical results that the Strang splitting scheme defined in (7.6) is second-order in time.

### 7.1.2. Peaceman-Rachford scheme

Now, we consider the Peaceman-Rachford scheme [57, 118] also for the 2D+time advection-diffusion equation. Here, we integrate the solution from $t_k$ to $t_{k+1},\ \forall k = 1, \dots, m$ in two different substeps as follows

$$\begin{cases} \dfrac{u^{k+1/2} - u^k}{\tau/2} + \mathcal{L}_1 u^{k+1/2} = f^{k+1/2} - \mathcal{L}_2 u^k, \\ \dfrac{u^{k+1} - u^{k+1/2}}{\tau/2} + \mathcal{L}_2 u^{k+1} = f^{k+1/2} - \mathcal{L}_1 u^{k+1/2}. \end{cases} \tag{7.7}$$

We derive a variational formulation of scheme (7.7) as

$$\begin{cases} \left(u^{k+1/2}, v\right) + \dfrac{\tau}{2}\left(\nu \dfrac{\partial u^{k+1/2}}{\partial x}, \dfrac{\partial v}{\partial x}\right) + \dfrac{\tau}{2}\left(\beta_x \dfrac{\partial u^{k+1/2}}{\partial x}, v\right) \\ \qquad = \left(u^k, v\right) - \dfrac{\tau}{2}\left(\nu \dfrac{\partial u^k}{\partial y}, \dfrac{\partial v}{\partial y}\right) - \dfrac{\tau}{2}\left(\beta_y \dfrac{\partial u^k}{\partial y}, v\right) \\ \qquad + \dfrac{\tau}{2}\left(f^{k+1/2}, v\right), \\ \\ \left(u^{k+1}, v\right) + \dfrac{\tau}{2}\left(\nu \dfrac{\partial u^{k+1}}{\partial y}, \dfrac{\partial v}{\partial y}\right) + \dfrac{\tau}{2}\left(\beta_y \dfrac{\partial u^{k+1}}{\partial y}, v\right) \\ \qquad = \left(u^{k+1/2}, v\right) - \dfrac{\tau}{2}\left(\nu \dfrac{\partial u^{k+1/2}}{\partial x}, \dfrac{\partial v}{\partial x}\right) - \dfrac{\tau}{2}\left(\beta_x \dfrac{\partial u^{k+1/2}}{\partial x}, v\right) \\ \qquad + \dfrac{\tau}{2}\left(f^{k+1/2}, v\right), \end{cases} \tag{7.8}$$

and employing approximation (7.2a), we express problem (7.8) in matrix form as

$$\begin{cases} \left[M_p^x + \dfrac{\tau}{2}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u_h^{k+1/2} \\ \qquad = M_p^x \otimes \left[M_p^y - \dfrac{\tau}{2}\left(K_p^y + R_p^y\right)\right] u_h^k + \dfrac{\tau}{2} F^{k+1/2}, \\ M_p^x \otimes \left[M_p^y + \dfrac{\tau}{2}\left(K_p^y + R_p^y\right)\right] u_h^{k+1} \\ \qquad = \left[M_p^x - \dfrac{\tau}{2}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y u_h^{k+1/2} + \dfrac{\tau}{2} F^{k+1/2}. \end{cases} \tag{7.9}$$

It is well known that the Peaceman-Rachford scheme (7.7) is second order in time and unconditionally stable for diffusion problems. However, the generalization of this scheme to 3D does not preserve these properties [135]. Therefore, we consider another ADI scheme for the 3D case in the next section.

### 7.1.3. Douglas-Gunn scheme

In 3D+time, we consider the Douglas-Gunn scheme [55, 56] that is also second order in time and unconditionally stable for diffusion problems. Here, we integrate the solution from time step $t_k$ to $t_{k+1}$, $\forall k = 1, \dots, m$, in three substeps as follows

$$\begin{cases} \left(1 + \frac{\tau}{2}\mathcal{L}_1\right) u^{k+1/3} = \tau f^{k+1/2} + \left(1 - \frac{\tau}{2}\mathcal{L}_1 - \tau\mathcal{L}_2 - \tau\mathcal{L}_3\right) u^k, \\ \left(1 + \frac{\tau}{2}\mathcal{L}_2\right) u^{k+2/3} = u^{k+1/3} + \frac{\tau}{2}\mathcal{L}_2 u^k, \\ \left(1 + \frac{\tau}{2}\mathcal{L}_3\right) u^{k+1} = u^{k+2/3} + \frac{\tau}{2}\mathcal{L}_3 u^k. \end{cases} \tag{7.10}$$

Similarly as in (7.8), we derive a variational formulation of scheme (7.10) as

$$\begin{cases} \left(u^{k+1/3}, v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u^{k+1/3}}{\partial x}, \frac{\partial v}{\partial x}\right) + \frac{\tau}{2}\left(\beta_x\frac{\partial u^{n+1/3}}{\partial x}, v\right) \\ \qquad = \left(u^k, v\right) - \frac{\tau}{2}\left(\nu\frac{\partial u^k}{\partial x}, \frac{\partial v}{\partial x}\right) - \frac{\tau}{2}\left(\beta_x\frac{\partial u^k}{\partial x}, v\right) \\ \qquad - \tau\left(\nu\frac{\partial u^k}{\partial y}, \frac{\partial v}{\partial y}\right) - \tau\left(\beta_y\frac{\partial u^k}{\partial y}, v\right) \\ \qquad - \tau\left(\nu\frac{\partial u^k}{\partial z}, \frac{\partial v}{\partial z}\right) - \tau\left(\beta_z\frac{\partial u^k}{\partial z}, v\right) + \tau\left(f^{k+1/2}, v\right), \\ \left(u^{k+2/3}, v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u^{k+2/3}}{\partial y}, \frac{\partial v}{\partial y}\right) + \frac{\tau}{2}\left(\beta_y\frac{\partial u^{k+2/3}}{\partial y}, v\right) \\ \qquad = \left(u^{k+1/3}, v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u^k}{\partial y}, \frac{\partial v}{\partial y}\right) + \frac{\tau}{2}\left(\beta_y\frac{\partial u^k}{\partial y}, v\right), \\ \left(u^{k+1}, v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u^{k+1}}{\partial z}, \frac{\partial v}{\partial z}\right) + \frac{\tau}{2}\left(\beta_z\frac{\partial u^{n+1}}{\partial z}, v\right) \\ \qquad = \left(u^{k+2/3}, v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u^k}{\partial z}, \frac{\partial v}{\partial z}\right) + \frac{\tau}{2}\left(\beta_z\frac{\partial u^k}{\partial z}, v\right). \end{cases} \tag{7.11}$$

Finally, employing (7.2b) and expressing problem (7.11) in matrix form we

obtain

$$\begin{cases} \left[M_p^x + \frac{\tau}{2}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y \otimes M_p^z u_h^{k+1/3} \\ \qquad = \left[M_p^x - \frac{\tau}{2}\left(K_p^x + R_p^x\right)\right] \otimes M_p^y \otimes M_p^z u_h^k \\ \qquad - \tau M_p^x \otimes \left(K_p^y + R_p^y\right) \otimes M_p^z u_h^k \\ \qquad - \tau M_p^x \otimes M_p^y \otimes \left(K_p^z + R_p^z\right) u_h^k + \tau F^{k+1/2}, \\ \\ M_p^x \otimes \left[M_p^y + \frac{\tau}{2}\left(K_p^y + R_p^y\right)\right] \otimes M_p^z u_h^{k+2/3} \\ \qquad = M_p^x \otimes M_p^y \otimes M_p^z u^{k+1/3} + M_p^x \otimes \frac{\tau}{2}\left(K_p^y + R_p^y\right) \otimes M_p^z u_h^k, \\ \\ M_p^x \otimes M_p^y \otimes \left[M_p^z + \frac{\tau}{2}\left(K_p^z + R_p^z\right)\right] u^{k+1} \\ \qquad = M_p^x \otimes M_p^y \otimes M_p^z u_h^{k+2/3} + M_p^x \otimes M_p^y \otimes \frac{\tau}{2}\left(K_p^z + R_p^z\right) u_h^k, \end{cases} \tag{7.12}$$

where $M_p^{x,y,z}$, $K_p^{x,y,z}$ and $R_p^{x,y,z}$ are the 1D mass, stiffness, and advection matrices of order $p$, respectively.

Note that in each time step of schemes (7.6), (7.9), and (7.12) we have Kronecker product of 1D matrices in the left-hand-side. Therefore, we can solve both schemes in linear computational cost with a direct solver [99, 100] (See Appendix C.1 for details).

## 7.2. iGRM method

In this section, we consider the Peaceman-Rachford (7.8) and the Douglas-Gunn (7.11) schemes. The extension to the Strang splitting scheme (7.5) is straightforward. In (7.8) and (7.11), at every time step we solve a problem of the form

$$\begin{cases} \text{Find } u \in U \text{ such that} \\ b(u,v) = l(w,v), \ \forall v \in V, \end{cases} \tag{7.13}$$

where $U$ and $V$ are Hilbert spaces (not necessarily equal) and $w \in U$ is known from the previous substep.

In particular, we have in both schemes

$$b\left(u,v\right) = \left(u,v\right) + \frac{\tau}{2}\left(\nu\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i}\right) + \frac{\tau}{2}\left(\beta_{x_i}\frac{\partial u}{\partial x_i}, v\right), \tag{7.14}$$

where $i \in \{1,2\}$ in (7.8) and $i \in \{1,2,3\}$ in (7.11), so we have denoted by $\{x_1, x_2, x_3\} := \{x, y, z\}$.

The right-hand-side of (7.13) for the Peaceman-Rachford scheme (7.8) is

$$l\,(w,v) = (w,v) - \frac{\tau}{2}\left(\nu\frac{\partial w}{\partial x_i}, \frac{\partial v}{\partial x_i}\right) - \frac{\tau}{2}\left(\beta_{x_i}\frac{\partial w}{\partial x_i}, v\right) + \frac{\tau}{2}(f,v), \tag{7.15}$$

with $i \in \{1,2\}$. For the Douglas-Gunn scheme (7.8), we have the following right-hand-side for the first substep

$$\begin{aligned} l\,(w,v) = (w,v) &- \frac{\tau}{2}\left(\nu\frac{\partial w}{\partial x_1}, \frac{\partial v}{\partial x_1}\right) - \frac{\tau}{2}\left(\beta_{x_1}\frac{\partial w}{\partial x_1}, v\right) \\ &- \tau\left(\nu\frac{\partial w}{\partial x_2}, \frac{\partial v}{\partial x_2}\right) - \tau\left(\beta_{x_2}\frac{\partial w}{\partial x_2}, v\right) \\ &- \tau\left(\nu\frac{\partial w}{\partial x_3}, \frac{\partial v}{\partial x_3}\right) - \tau\left(\beta_{x_3}\frac{\partial w}{\partial x_3}, v\right) + \tau(f,v), \end{aligned} \tag{7.16}$$

and for $i \in \{2,3\}$ we have

$$l\,(w,v) = (w,v) - \frac{\tau}{2}\left(\nu\frac{\partial w}{\partial x_i}, \frac{\partial v}{\partial x_i}\right) - \frac{\tau}{2}\left(\beta_{x_i}\frac{\partial w}{\partial x_i}, v\right). \tag{7.17}$$

In order to achieve stability in the schemes defined in Section 7.1, we apply a residual minimization method as defined in [47] to each time step. Then, our problem reduces to the following semi-infinite problem at each time step

$$\begin{cases} \text{Find } \{r, u_h\} \in V \times U_h \text{ such that} \\ (r,v)_V - b(u_h, v) = -l(v),\ \forall v \in V, \\ \qquad\qquad b(w_h, r) = 0,\ \forall w_h \in U_h, \end{cases} \tag{7.18}$$

where $U_h \subset U$ and $r$ is called the error representation function (See Appendix C.2 for details). Here, we have omitted symbol $w$ in $l(w,v)$, which refers to the information coming from the previous time substep to avoid confusion with the test functions $w_h \in U_h$.

Now, we discretize the test space $V_n \subset V$ and we obtain a fully discrete problem per time substep

$$\begin{cases} \text{Find } \{r_n, u_h\} \in V_n \times U_h \text{ such that} \\ (r_n, v_n)_{V_n} - b(u_h, v_n) = -l(v_n),\ \forall v_n \in V_n, \\ \qquad\qquad b(w_h, r_n) = 0,\ \forall w_h \in U_h. \end{cases} \tag{7.19}$$

Finally, in order to preserve the Kronecker product in problem (7.19), we seek the solution in the following space

$$U = V = \left\{ v \mid \int_\Omega \left( v^2 + \frac{\partial v}{\partial x_i} \frac{\partial v}{\partial x_i} \right) < +\infty \right\}, \tag{7.20}$$

with inner product defined as

$$(u, v)_V = (u, v) + \left( \frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i} \right), \tag{7.21}$$

where $i \in \{1, 2, 3\}$ depending on the first, second, or third substep of the ADI method.

*Remark* 7.1. In problem (7.19), we define the discrete test space $V_n$ in such a way that it is as close as possible to the abstract $V$ space, to ensure stability in a sense that the discrete inf-sup condition is satisfied. In our method it is possible to gain stability enriching the test space $V_n$ without changing the trial space $U_h$.

Taking into account the definitions of bilinear form (7.14), right-hand-sides (7.15)-(7.17), and the inner product (7.21), method (7.19) is defined as

$$\begin{cases} \text{Find } \{r_n, u_h\} \in V_n \times U_h \text{ such that} \\ (r_n, v_n)_{V_n} - (u_h, v_n) - \dfrac{\tau}{2} \left( \nu \dfrac{\partial u_h}{\partial x_i}, \dfrac{\partial v_n}{\partial x_i} \right) - \dfrac{\tau}{2} \left( \beta_{x_i} \dfrac{\partial u_h}{\partial x_i}, v_n \right) = -l(v_n), \ \forall v_n \in V_n, \\ (w_h, r_n) + \dfrac{\tau}{2} \left( \nu \dfrac{\partial w_h}{\partial x_i}, \dfrac{\partial r_n}{\partial x_i} \right) + \dfrac{\tau}{2} \left( \beta_{x_i} \dfrac{\partial w_h}{\partial x_i}, r_n \right) = 0, \ \forall w_h \in U_h. \end{cases} \tag{7.22}$$

Now, we explain how we select the discrete spaces in (7.22) for the $2D$ Peaceman-Rachford scheme (7.8). The extension to 3D using the Douglas-Gunn scheme (7.11) is straightforward.

First, we approximate the solution as a tensor product of one dimensional B-splines basis functions of order $p$ as in (7.2a). Then, we test with a tensor product of one dimensional B-splines basis functions, where we enrich the order from $p$ to $s$ (with $s \geqslant p$) in the direction of the alternating splitting

$$v_n = \begin{cases} B_{o;s}(x) B_{q;p}(y), \ \text{(substep 1)}, \\ B_{o;p}(x) B_{q;s}(y), \ \text{(substep 2)}. \end{cases}$$

Similarly, we approximate the residual enriching the order in the direction of the splitting as

$$r_n = \begin{cases} \displaystyle\sum_{l,m} r_{l,m} B_{l;s}(x) B_{m;p}(y), \ \text{(substep 1)}, \\ \displaystyle\sum_{l,m} r_{l,m} B_{l;p}(x) B_{m;s}(y), \ \text{(substep 2)}, \end{cases}$$

and finally, we test again with B-splines of order $p$

$$w_h = B_{i;p}(x)B_{j;p}(y).$$

*Remark* 7.2. We perform the enrichment of the test space in the alternating direction manner. In this way, when we solve the problem with derivatives along the $x$ direction, we enrich the test space by increasing the B-splines order in the $x$ direction, but we keep the order of the B-splines along $y$ constant (same as in the trial space). By doing that, we preserve the Kronecker product structure of the matrix to ensure that we can apply the alternating direction solver.

The residual minimization system (7.22) is of the form

$$\begin{bmatrix} G & -B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} r_n \\ u_h \end{bmatrix} = \begin{bmatrix} -L \\ 0 \end{bmatrix}, \tag{7.23}$$

where, from the definition of the inner product (7.21), the entries of matrix $G$ for the first substep are

$$\begin{aligned} G_{l,m,o,q} &= (B_{l;s}(x)B_{m;p}(y), B_{o;s}(x)B_{q;p}(y)) + \left(\frac{\partial}{\partial x}B_{l;s}(x)B_{m;p}(y), \frac{\partial}{\partial x}B_{o;s}(x)B_{q;p}(y)\right) \\ &= \left[(B_{l;s}(x), B_{o;s}(x))_{\Omega_x} + \left(\frac{\partial}{\partial x}B_{l;s}(x), \frac{\partial}{\partial x}B_{o;s}(x)\right)_{\Omega_x}\right](B_{m;p}(y), B_{q;p}(y))_{\Omega_y}, \end{aligned}$$

and from the definition of the bilinear form (7.14), the entries of matrix $B$ for the first substep are

$$\begin{aligned} B_{i,j,o,q} &= (B_{i;p}(x)B_{j;p}(y), B_{o;s}(x)B_{q;p}(y)) + \frac{\tau}{2}\left(\nu\frac{\partial}{\partial x}B_{i;p}(x)B_{j;p}(y), \frac{\partial}{\partial x}B_{o;s}(x)B_{q;p}(y)\right) \\ &\quad + \frac{\tau}{2}\left(\beta_x\frac{\partial}{\partial x}B_{i;p}(x)B_{j;p}(y), B_{o;s}(x)B_{q;p}(y)\right) \\ &= \left[(B_{i;p}(x), B_{o;s}(x))_{\Omega_x} + \frac{\tau}{2}\left(\nu\frac{\partial}{\partial x}B_{i;p}(x), \frac{\partial}{\partial x}B_{o;s}(x)\right)_{\Omega_x}\right. \\ &\quad \left. + \frac{\tau}{2}\left(\beta_x\frac{\partial}{\partial x}B_{i;p}(x), B_{o;s}(x)\right)_{\Omega_x}\right](B_{j;p}(y), B_{q;p}(y))_{\Omega_y}. \end{aligned}$$

Here, $(\cdot,\cdot)_{\Omega_x}$ denotes the $L^2(\Omega_x)$ inner product and a similar definition holds for $(\cdot,\cdot)_{\Omega_y}$.

Then, we have

$$G = \begin{cases} (M_s^x + K_s^x) \otimes M_p^y, \text{ (substep 1)}, \\ M_p^x \otimes (M_s^y + K_s^y), \text{ (substep 2)}, \end{cases}$$
$$B = \begin{cases} \left[M_{ps}^x + \frac{\tau}{2}\left(K_{ps}^x + R_{ps}^x\right)\right] \otimes M_p^y, \text{ (substep 1)}, \\ M_p^x \otimes \left[M_{ps}^y + \frac{\tau}{2}\left(K_{ps}^y + R_{ps}^y\right)\right], \text{ (substep 2)}, \end{cases}$$

where $M_s^{x,y}$ are the $1D$ mass matrices of order $s$ and $M_{ps}^{x,y}$ are the non-square mass matrices formed with B-splines of orders $p$ and $s$ (similarly for $K_s^{x,y}$, $K_{ps}^{x,y}$, $R_s^{x,y}$ and $R_s^{x,y}$). Finally, matrix in (7.23) becomes

$$\begin{bmatrix} G & -B \\ B^T & 0 \end{bmatrix} = \begin{bmatrix} M_s^x + K_s^x & -\left[M_{ps}^x + \frac{\tau}{2}\left(K_{ps}^x + R_{ps}^x\right)\right] \\ \left[M_{ps}^x + \frac{\tau}{2}\left(K_{ps}^x + R_{ps}^x\right)\right]^T & 0 \end{bmatrix} \otimes M_p^y,$$
$$\begin{bmatrix} G & -B \\ B^T & 0 \end{bmatrix} = M_p^x \otimes \begin{bmatrix} M_s^y + K_s^y & -\left[M_{ps}^y + \frac{\tau}{2}\left(K_{ps}^y + R_{ps}^y\right)\right] \\ \left[M_{ps}^y + \frac{\tau}{2}\left(K_{ps}^y + R_{ps}^y\right)\right]^T & 0 \end{bmatrix}, \tag{7.24}$$

where we have employed that $M_p^{x,y}$ are symmetric matrices and the following property of the Kronecker product

$$(A \otimes B)^T = A^T \otimes B^T.$$

The matrices in (7.24) have multi-diagonal structure and can be factorized in a linear $\mathcal{O}(N)$ computational cost.

## 7.3. Numerical results

### 7.3.1. Manufactured solution problem

In order to verify the convergence order of the time-integration schemes (7.6), (7.9) and (7.12), we build a time-depedent advection-diffusion problem with a manufactured solution. Namely, we consider (2.1) in 2D and 3D with

$$\begin{cases} 2D: \ \nu = 10^{-2}, \ \beta = [1\ 0], \ \Omega = [0,1]^2, \ I = (0,2], \\ 3D: \ \nu = 10^{-2}, \ \beta = [1\ 0\ 0], \ \Omega = [0,1]^3, \ I = (0,2], \end{cases}$$

and homogeneous Dirichlet boundary conditions. We setup the forcing function $f$ in such a way that it delivers the manufactured solution of the form

$$\begin{cases} u(x,y,t) = \sin(\pi x)\sin(\pi y)\sin(\pi t), \text{ in 2D}, \\ u(x,y,z,t) = \sin(\pi x)\sin(\pi y)\sin(\pi z)\sin(\pi t), \text{ in 3D}. \end{cases}$$

We employ iGRM method on a mesh of $2^5$ elements in each spatial direction with different number of time steps. We solve the problem with the Strang+CN (7.6), Peaceman-Rachford (7.9) and Douglas-Gunn (7.12) time integration schemes and the direction splitting solver using the Kronecker product structure of the matrices. We select (2,1) B-splines for trial and (2,0) for test. Here, we denote by $(p, r)$ the B-splines of order $p$ with $C^r$ continuity.

We compute the relative error in $L^2$ between the exact solution $u$ and the numerical solution $u_h$, i.e.,

$$\frac{||u(t) - u_h(t)||_{L^2}}{||u(t)||_{L^2}} \cdot 100.$$

Figure 7.2 shows the relative error for different time step sizes. We conclude that all schemes are second order accurate in time down to the spatial accuracy that is of $10^{-5}$ for $2^5$ elements in each direction.

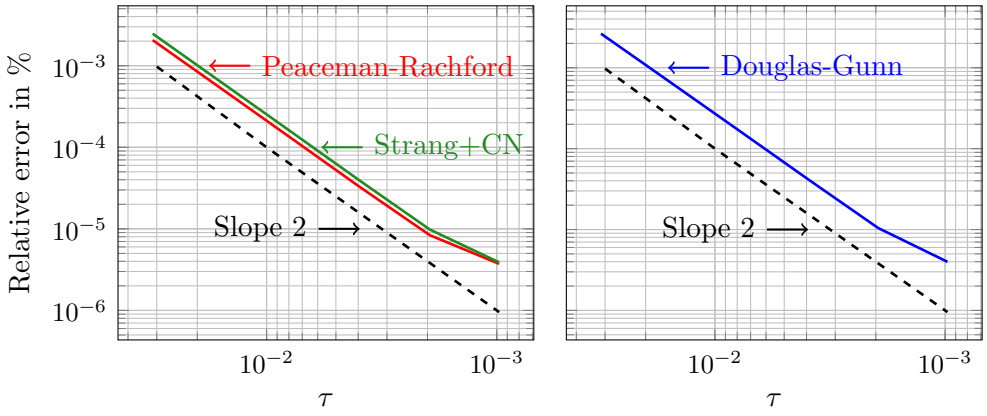



Figure 7.2.: Temporal convergence in $L^2$ norm for Strang+CN (2D), Peaceman-Rachford (2D) (left) and Douglas-Gunn (3D) (right) schemes on a mesh with $2^5$ elements in each spatial direction from [97, 98].

### 7.3.2. Circular wind problem

We consider the circular wind problem presented in [97], where the advection coefficient does not have Kronecker product structure. Namely, we consider problem (2.1) over a squared domain $\Omega = [-1, 1]^2$ with $\nu = 10^{-6}$, $\beta = [y \ -x]$, $f = 0$ and the initial condition $u_0$ as in the first snapshot of Figure 7.3.

We formulate the residual minimization system according to (7.19) with the Peaceman-Rachford scheme (7.9). In this case, we cannot employ the linear computational cost direction splitting solver, but we call MUMPS [5, 6] direct

solver to factorize the residual minimization system at each time step. We select a time step size of $\tau = 0.1$ and to gain stability we employ a mesh in space of $2^5 \times 2^5$ elements with $(4, 3)$ B-splines for trial and $(5, 0)$ for test. Figure 7.3 shows different snapshots of the solution.

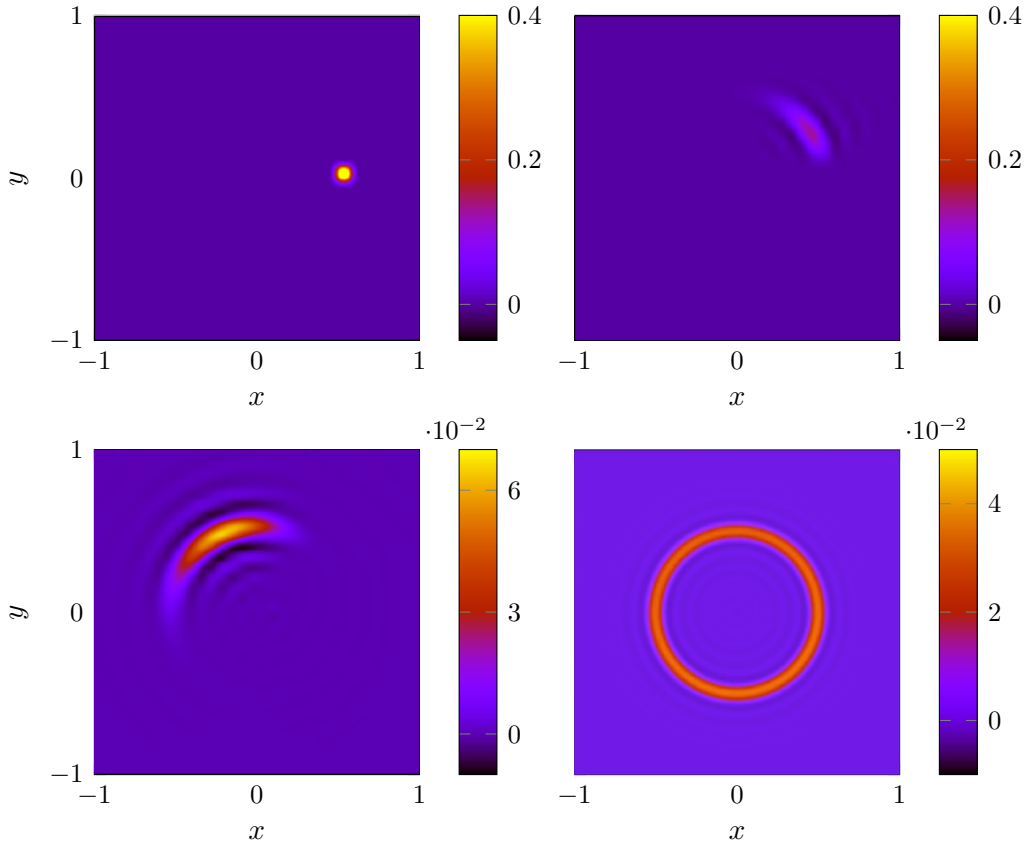



Figure 7.3.: Snapshots of the initial condition and time steps 6, 20 and 700 of the solution of the circular wind problem from [97].

Finally, since the computational cost is no longer linear $\mathcal{O}(N)$, we provide in Table 1 the execution times for MUMPS solver called for different configurations of mesh dimensions and trial and test spaces.

We conclude that, in this strongly advective problem where the Péclet number is $Pe = 10^6$, we obtain a stable solution with the iGRM method presented in this chapter.

| Trial | Test | Elements | DoF | Solver [ms] |
|---|---|---|---|---|
| (2,1) | (3,0) | 8 | 725 | 5 |
| (2,1) | (3,0) | 16 | 2725 | 27 |
| (2,1) | (3,0) | 32 | 10565 | 259 |
| (2,1) | (3,0) | 64 | 41605 | 1393 |
| (3,2) | (4,0) | 8 | 1210 | 14 |
| (3,2) | (4,0) | 16 | 4586 | 100 |
| (3,2) | (4,0) | 32 | 17866 | 928 |
| (3,2) | (4,0) | 64 | 70538 | 5725 |
| (4,3) | (5,0) | 8 | 1825 | 51 |
| (4,3) | (5,0) | 16 | 6961 | 618 |
| (4,3) | (5,0) | 32 | 27217 | 2722 |
| (4,3) | (5,0) | 64 | 107665 | 9116 |

Table 7.1.: Execution times reported by MUMPS solver for different configurations of mesh dimensions and trial and test spaces from [97].

### 7.3.3. Pollution propagation problem

In this section, we consider the advection-diffusion problem (2.1) over a 3D cube shape domain with dimensions $5000 \times 5000 \times 5000$ meters presented in [98]. We consider the diffusion diagonal tensor $\nu = \text{diag}[50\ 50\ 0.5]$ and the velocity field $\beta$ is given by

$$\beta = [\cos a(t)\ \ \sin a(t)\ v(t)],$$

where

$$a(t) = \frac{\pi}{3}(\sin(s(t)) + \frac{1}{2}\sin(2.3s(t))) + \frac{3}{8}\pi, \quad v(t) = \frac{1}{3}\sin(s(t)),$$

and $s(t) = \frac{t}{150}$. The source is defined as

$$f(p) = (r(p) - 1)^2(r(p) + 1)^2,$$

where $r(p) = \min(1, (|p - p_0|/25)^2)$, $p$ represents the distance from the source, and $p_0$ is the location of the source that in this case is $p_0 = (3, 3, 2)$. The initial state is defined as the constant concentration of order $10^{-6}$ in the entire domain. In this problem, we model the propagation of the pollutant generated by a single

source $f$, distributed by the wind blowing with changing directions in time $\beta$ and the diffusion phenomena $\nu$.

For the numerical results, we consider a mesh with $50 \times 50 \times 50$ elements, we select $\tau = 0.1$ and we perform 300 time steps of the Douglas-Gunn scheme (7.12). First, we solve the problem with the standard Galerkin formulation with (2,1) B-splines for both trial and test. We perform direction splitting but without the residual minimization method. Figure 7.4 shows the snapshots of time steps 30, 60, 90, 120, 150 and 180.

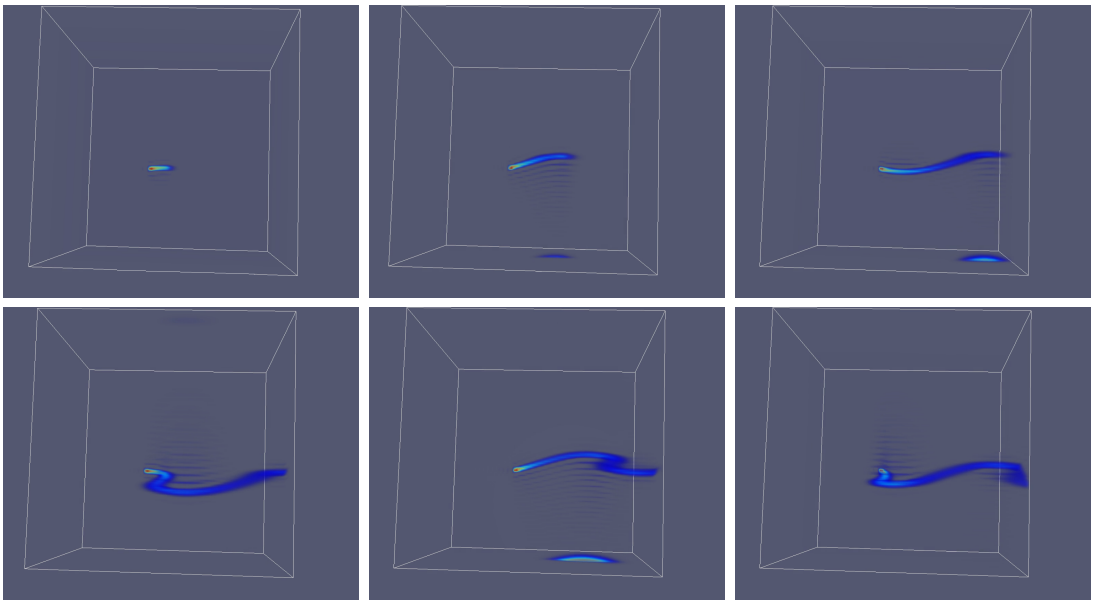


Figure 7.4.: Snapshots of time steps 30, 60, 90, 120, 150 and 180 of the Galerkin method from [98].

In the snapshots of Figure 7.4 we observe "oscillations" and "reflections". Since the simulation is supposed to model the propagation of the pollutant from a chimney by means of the advection (wind) and diffusion phenomena, the oscillations and reflections on the boundaries are unexpected. Both these phenomena appear and disappear during the entire simulation and they do not cause a blowup but unexpected local behavior.

To improve the spatial stability of the simulation, we add now the residual minimization method at each time step. We consider a mesh of $50 \times 50 \times 50$ elements, with $(2, 1)$ B-splines for trial and $(3, 2)$ for test. We perform 300 time steps and we present the snapshots in time steps 30, 60, 90, 120, 150 and 180 in Figure 7.5. We conclude that with the iGRM method we do not observe the "oscillations" and "reflections" presented in the Galerkin simulation for the 3D+time advection-diffusion equation. Therefore, we obtain a more stable solution.

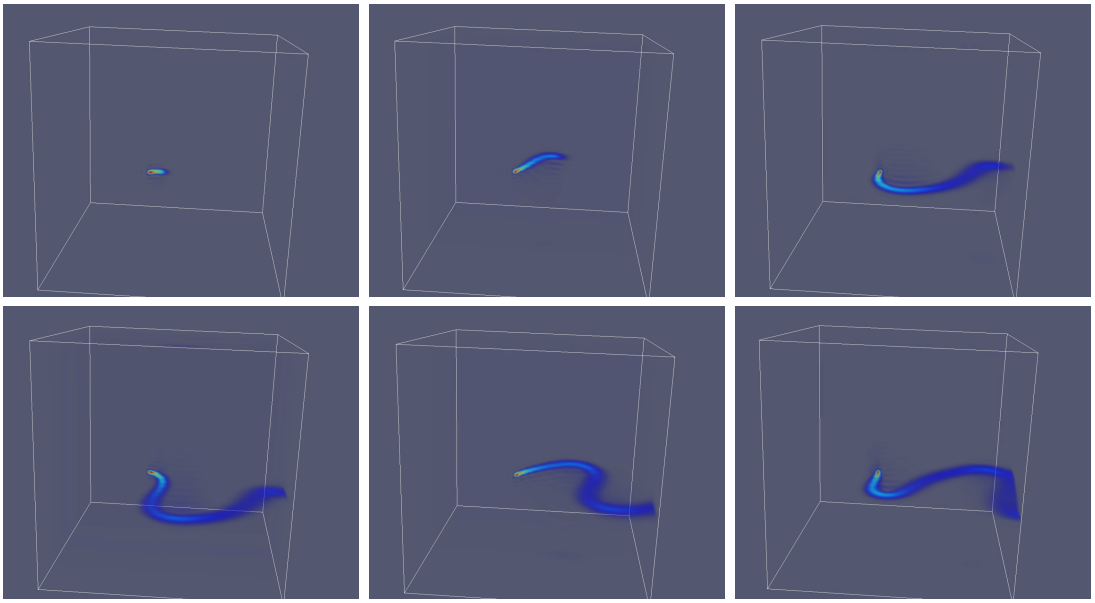


Figure 7.5.: Snapshots of time steps 30, 60, 90, 120, 150 and 180 of the iGRM method from [98].

In conclusion, we propose a novel computational implicit method, which we call iGRM with direction splitting, that mixes the benefits resulting from IGA, residual minimization, and ADI schemes. We show that our method is stable for time-dependent advection-dominated-diffusion problems.

We will study how to derive a goal-oriented adaptive strategy employing iGRM with direction splitting as a future work. One option is to follow the ideas from [48, 92] and for each time step of the dual problem solve another saddle-point problem like (7.23) with the output functional in the right-hand-side. Then, we would obtain two error representation functions: one coming from the direct problem and the other one from the dual problem. Finally, we can employ both errors to perform refinements in a goal-oriented approach.

# 8. Conclusions and future work

## 8.1. Conclusions

In this dissertation, we first propose a discontinuous-in-time Petrov-Galerkin (dPG) formulation of the linear diffusion equation that, after exact integration in time, leads to explicit Runge-Kutta (RK) methods. We define families of piecewise polynomials for trial and test functions that give rise to any stage RK method and we define the trial and test spaces as the span of those families. We characterize all the second-order and two-stage explicit RK methods and we provide explicit examples up to four stages. When the trial functions are polynomials of order $p$ in time, then the test space is formed by incomplete polynomial spaces of order $p+1$. Alternatively, we can define the test space to be a complete polynomial space of order p and the resulting trial space will contain some polynomials of order $p+1$, but in this case we loose approximability of the solution. We present a constructive method that, given a Butcher table, systematically builds trial and test functions corresponding to explicit RK methods in time. A limitation of our method is that, for a large number of stages, we end up with large nonlinear systems of equations that are difficult to solve.

Employing the presented dPG formulation, we can naturally represent the error in the Quantity of Interest (QoI). Therefore, this variational structure reproduces the existing goal-oriented space-time adaptive algorithms but employing explicit-in-time RK schemes. In this work, we also propose an explicit-in-time goal-oriented adaptive algorithm for the linear advection-diffusion equation. We select piecewise constant trial and test functions in time to derive the Forward Euler method for the primal problem. Interchanging the trial and test spaces, we also derive the Forward Euler method for the dual problem but running backwards in time. Then, the error in the QoI is expressed employing the errors in space of the primal and dual problems. The adaptive algorithm we propose performs local goal-oriented adaptive refinements in space. The time grid is adapted locally (for the entire space) based on the Courant-Friedrichs-Lewy (CFL) condition in order to ensure the stability of the method. We illustrate the performance of the algorithm in one-dimensional diffusion and advection-diffusion problems.

In this dissertation, we also propose a forward-in-time goal-oriented adaptive algorithm. We define a pseudo-dual problem that, as the primal problem, runs

forwards in time. Then, we derive an error representation employing the error of the primal problem and the solution of the pseudo-dual problem. The goal-oriented adaptive algorithm we present performs local refinements in space for a fixed time grid. We compare the proposed algorithm with the classical one for one-dimensional advection-diffusion problems. We conclude that our algorithm only performs properly for diffusion problems where the output functional has support over the whole time interval. For advection-diffusion problems, other strategies need to be considered. Herein, we propose a hybrid algorithm, where we first compute an initial representation of the adjoint solution on the entire domain and then use this partial information to guide the refinement process forwards in time.

Employing similar ideas, we propose an error representation that uses (unconventional) pseudo-dual problems for Goal-Oriented Adaptivity (GOA) in time-domain wave propagation problems. We build a pseudo-dual problem by modifying the standard adjoint problem in a way that we obtain another wave propagation problem with better stability properties. We apply this new error representation to the one-dimensional wave equation and we compare the classical upper bounds of the error in the QoI with the proposed ones. We conclude that, for the 1D wave equation, the new bounds are sharper than the classical ones and therefore, we obtain a better criteria to perform the adaptive process.

Finally, we present a stabilized Isogeometric Analysis (IGA)-Finite Element Method (FEM) with Alternating Direction Implicit (ADI) schemes for time-dependet advection-diffusion problem. The application of B-spline basis functions for the approximation of the numerical solutions results in a smooth, higher order approximation of the solution. We apply a minimum residual method at each time step and we keep the Kronecker product structure of the matrix, resulting in a linear computational cost of the direct solver. We called our method iGRM with direction splitting, whose performance has been analyzed with 2D and 3D+time advection-dominated-difusion problems.

## 8.2. Future work

The dPG formulation presented in Chapter 3 could be employed to build new time-stepping schemes of RK type or more general ones, and also to extend the existing space discretizations like IGA [69, 81, 144] or Discontinuous Petrov-Galerkin (DPG) [47] to time domain problems. As future work, we plan to analyze the stability of the new time marching schemes arising from our dPG construction and explore the ideas from the DPG community in order to build more stable explicit methods. The presented formulation could be useful to study the variational structure of other implicit and explicit methods such as Adams-

Bashforth, Adams-Moulton or Backward Differentiation Formulas (BDF).

A possible extension of the work presented in Chapter 4 is to perform GOA employing higher-order explicit time-marching schemes, as those shown in Chapter 3. Also, for the cases in which verification of the CFL may be insufficient to control the time-discretization error, it may be necessary to consider a more sophisticated goal-oriented algorithm intended to simultaneously control both the space and time errors. The presented space-time variational formulation allows to perform local space-time (goal-oriented) refinements. We will study this possibility in the future and we will also develop goal-oriented adaptive algorithms for higher-dimensionality problems in space.

Regarding the work presented in Chapters 5 and 6, a possible extension is the application of the presented ideas to 2D and 3D+time problems. Moreover, in Chapter 5 we can study other alternative bilinear forms preserving the Galerkin orthogonality property in order to obtain sharper upper bounds of the error but without loosing the pseudo-dual problem running forwards in time.

Finally, we will extend the method defined in Chapter 7 to more complex equations and geometries. We will also derive a goal-oriented adaptive strategy following the ideas presented in [48, 92]. We will apply the iGRM with direction splitting to both primal and dual problems and then employ both error representation functions as indicators to perform adaptivity.

# 9. Main achievements

## 9.1. Peer-reviewed publications

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Forward-in-time goal-oriented adaptivity.*** International Journal for Numerical Methods in Engineering, 2019, 1-16. https://doi.org/10.1002/nme.6059.

**2019** J. Muñoz-Matute, V. M. Calo, D. Pardo, E. Alberdi and K. G. van der Zee. ***Explicit-in-time goal-oriented adaptivity.*** Computer Methods in Applied Mechanics and Engineering, 2019, vol. 347, p. 176-200. https://doi.org/10.1016/j.cma.2018.12.028.

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Variational formulations for explicit Runge-Kutta methods.*** Accepted in Finite Elements in Analysis & Design, June 2019.

**2019** M. Łoś, J. Muñoz-Matute, I. Muga and M. Paszyński. ***Isogeometric residual minimization method (iGRM) with direction splitting for implicit problems.*** Accepted in Computers & Mathematics with Applications, June 2019.

**2019** M. Łoś, J. Muñoz-Matute, K. Pingali, I. Muga and M. Paszyński. ***Parallel shared-memory isogeometric residual minimization (iGRM) simulations of 3D advection-diffusion problems.*** Submitted to Engineering with Computers, April 2019.

**2017** J. Muñoz-Matute, E. Alberdi, D. Pardo and V. M. Calo. ***Time-domain goal-oriented adaptivity using pseudo-dual error representations.*** Computer Methods in Applied Mechanics and Engineering, 2017, vol. 325, p. 395-415. https://doi.org/10.1016/j.cma.2017.06.037.

## 9.2. International conferences

**2019** J. Muñoz-Matute, M. Łoś, I. Muga and M. Paszyński. ***Isogeometric residual minimization method (iGRM) with direction splitting***

***for time-dependent advection-diffusion problems.***
ICCS 2019, Faro, Portugal.

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Explicit time integrators and forward-in-time goal oriented adaptivity.*** ADMOS 2019, Alicante, Spain.

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Variational formulation and error estimation for explicit time integrators.*** SIAM CSE19, Spokane, Washington, USA.

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Time discontinuous Petrov-Galerkin methods for the advection-diffusion equation.*** WONAPDE 2019, Concepción, Chile.

**2018** J. Muñoz-Matute, V. M. Calo, D. Pardo and E. Alberdi. ***Error representation and space-time goal-oriented adaptivity for the advection-diffusion equation employing explicit Runge-Kutta methods.*** WCCM 2018, New York, USA.

**2018** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Space-time goal-oriented adaptivity and error estimation for parabolic problems employing explicit Runge-Kutta methods.*** ICCS 2018, Wuxi, China.

**2016** J. Muñoz-Matute, E. Alberdi, D. Pardo and Á. Rodríguez-Rozas. ***Time-domain goal-oriented adaptivity using unconventional error representations.*** ICCS 2016, San Diego, USA.

## 9.3. Seminars & Workshops

**2019** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Goal-oriented adaptivity employing forward-in-time pseudo-dual problems.*** Workshop: "Valparaíso's Mathematics and its Applications Days (V-MAD 9)", Valparaíso, Chile.

**2018** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Goal-oriented adaptivity for the advection-diffusion equation employing explicit time integrators.*** BCAM LIGHT Seminar, Bilbao, Spain.

**2018** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Galerkin methods for time-domain goal-oriented adaptivity.*** Workshop: "Fifth

International Congress on Multiphysics, Multiscale and Optimization Problems $(m^2op)^V$", Bilbao, Spain.

**2018** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Discontinuous-in-time explicit Petrov-Galerkin formulations for parabolic problems.*** Seminar at AGH University of Science and Technology, Krakow, Poland.

**2018** J. Muñoz-Matute, D. Pardo, V. M. Calo and E. Alberdi. ***Time-domain goal-oriented adaptivity employing explicit Galerkin time integrators.*** Seminar at the University of Nottingham, England, UK.

**2017** E. Alberdi, V. Darrigrand, J. Muñoz, D. Pardo, V. M. Calo, Á. Rodríguez-Rozas and I. Muga. ***Pseudo-dual error representations for goal-oriented adaptivity: applications to time-domain and Helmholtz problems.*** Workshop: "Geophysical Applications and HPC", Barcelona, Spain.

## 9.4. Research stays

**2018-2019** AGH University of Science and Technology, Krakow (Poland)
**Supervisor:** Maciej Paszyński
**Dates:** 4 - 14 April 2018 / 8 - 20 September 2019 (3.5 weeks)

**2019** Oden Institute for Computational Engineering and Sciences,
The University of Texas at Austin, Austin, Texas (USA)
**Supervisor:** Leszek F. Demkowicz
**Dates:** 7 July - 16 August 2019 (1.5 months)

**2018-2019** Pontificia Universidad Católica de Valparaíso, Valparaíso (Chile)
**Supervisor:** Ignacio Muga
**Dates:** 25 December 2018 - 4 February 2019 (1.5 months)

**2017-2018** Curtin University, Perth, WA (Australia)
**Supervisor:** Victor M. Calo
**Dates:** 1 March - 31 May 2017 / 28 September - 10 December 2017 / 21 September - 26 November 2018 (8 months)

**2018** The University of Nottingham, England (UK)
**Supervisor:** Kristoffer Van der Zee
**Dates:** 11 - 24 March 2018 (2 weeks)

# A. Nonlinear system

In this section, we express (3.26) in matrix form and we provide a MATLAB code to solve it for a general number of stages.

## A.1. Matrix form

We consider, for example, $s$-stages and trial and test functions of order $s$ over the master element $[0, 1]$

$$\begin{cases} \phi_1(t) = c_{10} + c_{11}t + \ldots + c_{1s}t^s, \\ \qquad \vdots \\ \phi_s(t) = c_{s0} + c_{s1}t + \ldots + c_{ss}t^s, \\ \varphi_1(t) = d_{10} + d_{11}t + \ldots + d_{1s}t^s, \\ \qquad \vdots \\ \varphi_s(t) = d_{s0} + d_{s1}t + \ldots + d_{ss}t^s. \end{cases}$$

To simplify notation, we collect the entries into the following matrices

$$C := \begin{bmatrix} c_{10} & c_{11} & \cdots & c_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{s0} & c_{s1} & \cdots & c_{ss} \end{bmatrix},$$

$$D := \begin{bmatrix} d_{10} & d_{11} & \cdots & d_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ d_{s0} & d_{s1} & \cdots & d_{ss} \end{bmatrix}.$$

Thus, we can write conditions (3.26) in matrix form as

$$\begin{cases} C\,\mathbf{e}_{1,s+1} = \mathbf{e}_{1,s}, \\ D\,\mathbf{1}_{s+1} = \mathbf{e}_{1,s}, \\ DAC^T - BC^T = E, \\ DFC^T = G, \end{cases}$$

where

$$\mathbf{e}_{1,s+1} := \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T, \quad \mathbf{1}_{s+1} := \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T,$$

$$A := \begin{bmatrix} 0 & 1 & \cdots & 1 & 1 \\ 0 & 1/2 & \cdots & (s-1)/s & s/(s+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1/s & \cdots & 1/2 & s/(2s-1) \\ 0 & 1/(s+1) & \cdots & (s-1)/(2s-1) & 1/2 \end{bmatrix},$$

$$B := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad E := \begin{bmatrix} -1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 1 \end{bmatrix},$$

$$F := \begin{bmatrix} 1 & 1/2 & \cdots & 1/s & 1/(s+1) \\ 1/2 & 1/3 & \cdots & 1/(s+1) & 1/(s+2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/s & 1/(s+1) & \cdots & 1/(2s-1) & 1/2s \\ 1/(s+1) & 1/(s+2) & \cdots & 1/2s & 1/(2s+1) \end{bmatrix},$$

$$G := \begin{bmatrix} b_1 & b_2 & \cdots & b_s \\ a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{ss} \end{bmatrix}.$$

*A. Nonlinear system*

We compute the entries in the matrices $A$ and $F$ from

$$A = \int_0^1 \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{s-1} \\ t^s \end{bmatrix} \begin{bmatrix} 0 & 1 & \cdots & (s-1)t^{s-2} & st^{s-1} \end{bmatrix} dt$$

$$= \int_0^1 \begin{bmatrix} 0 & 1 & \cdots & (s-1)t^{s-2} & st^{s-1} \\ 0 & t & \cdots & (s-1)t^{s-1} & st^s \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & t^{s-1} & \cdots & (s-1)t^{2s-3} & st^{2s-2} \\ 0 & t^s & \cdots & (s-1)t^{2s-2} & st^{2s-1} \end{bmatrix} dt$$

$$= \begin{bmatrix} 0 & 1 & \cdots & 1 & 1 \\ 0 & 1/2 & \cdots & (s-1)/s & s/(s+1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1/s & \cdots & 1/2 & s/(2s-1) \\ 0 & 1/(s+1) & \cdots & (s-1)/(2s-1) & 1/2 \end{bmatrix},$$

$$F = \int_0^1 \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{s-1} \\ t^s \end{bmatrix} \begin{bmatrix} 1 & t & \cdots & t^{s-1} & t^s \end{bmatrix} dt$$

$$= \begin{bmatrix} 1 & 1/2 & \cdots & 1/s & 1/(s+1) \\ 1/2 & 1/3 & \cdots & 1/(s+1) & 1/(s+2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/s & 1/(s+1) & \cdots & 1/(2s-1) & 1/2s \\ 1/(s+1) & 1/(s+2) & \cdots & 1/2s & 1/(2s+1) \end{bmatrix}.$$

## A.2. MATLAB Code

```
%Script to calculate the trial and test functions of Runge
    -Kutta methods
%Import data
[n_fun,p_trial,p_test,V_triv,V_deriv,V_grad]=data;
%Initialize the solution
S_trial=cell(n_fun,p_trial+1);
S_test=cell(n_fun,p_test+1);
%Write the coefficients
coef_trial=sym('c%d%d',[n_fun p_trial+1]);
coef_test=sym('d%d%d',[n_fun p_test+1]);
%Write the conditions
[C_triv,C_deriv,C_grad]=conditions(n_fun,coef_trial,
    coef_test,p_trial,p_test);
%Solve the nonlinear system
[S_trial{1:end},S_test{1:end}]=solve([C_triv C_deriv
    C_grad]==[V_triv V_deriv V_grad],[coef_trial coef_test
    ]);
```

```
function [n_fun,p_trial,p_test,V_triv,V_deriv,V_grad] =
    data()
%Function to write the data
%Number of trial and test functions and their order
n_fun=2;
p_trial=2;
p_test=2;
%The value of the conditions. For example:
%2-stage Runge-Kutta method
V_triv=[1 1;0 0];
V_deriv=[-1 0;-1 1];
alpha=1;
V_grad=[1-1/(2*alpha) 1/(2*alpha);alpha 0];
%Forward Euler method
%V_triv=[1 1];
%V_deriv=-1;
%V_grad=1;
end
```

```
function [C_triv,C_deriv,C_grad]= conditions(n_fun,
    coef_trial,coef_test,p_trial,p_test)
%Function to write the conditions in matrix form the trial
    and test functions must satisfy
C_trial=coef_trial*eye(p_trial+1,1);
C_test=coef_test*ones(p_test+1,1);
C_triv=[C_trial  C_test];

C=repmat((1:p_test+1)',1,p_trial+1)+repmat(0:p_trial,
    p_test+1,1);
C=1./C;
C_grad=coef_test*C*coef_trial.';

B=[ones(1,p_trial+1);zeros(n_fun-1,p_trial+1)];
A=repmat(0:p_trial,p_test+1,1).*[zeros(p_test+1,1) C(:,1:
    end-1)];
C_deriv=coef_test*A*coef_trial.'-B*coef_trial.';
end
```

# B. Stability analysis and definition of $B^{*-}_{DG}(\cdot,\cdot)$

In this appendix, we analyze the Courant-Friedrichs-Lewy (CFL) condition of the advection-diffusion equation when we employ the Forward Euler method in time and piecewise linear functions in space. We also explain how we derive $B^{*-}_{DG}(\cdot,\cdot)$ from $B^{-}_{DG}(\cdot,\cdot)$

## B.1. Stability analysis

The stability of problems (3.6) and (4.22) depends on whether the eigenvalues of the matrix $A := -M^{-1}(K+R)$ (where $M$, $K$ and $R$ are the mass, stiffness and weak derivative matrices, respectively) are included in the stability region of the time integration method. Figure B.1 shows the stability regions of the explicit Runge-Kutta (RK) methods of $s$ stages and order $p$, when $s = p$. The Forward Euler method is a RK method with $s = p = 1$ [73].

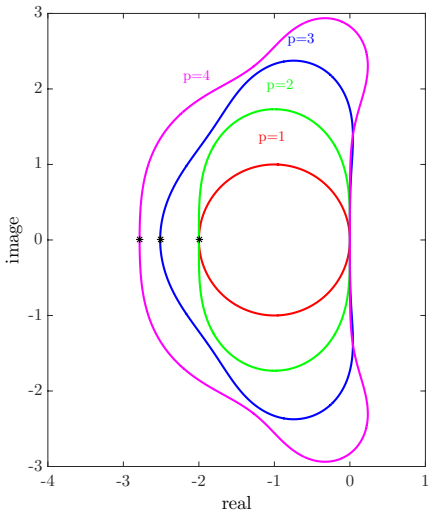



Figure B.1.: Stability of the explicit RK methods when $s = p$ (interior to curves).

Now, we derive the CFL condition for one space dimension. We assume that $\nu > 0$ and $\beta$ are constants. We consider uniform spatial and temporal meshes, where $h$ is the size of each element in space and $\tau$ is the time step size. If we employ piecewise linear basis functions in space, the elemental matrices are

$$M_{elem} = h \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}, \; K_{elem} = \frac{\nu}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \; R_{elem} = \beta \begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix}.$$

We need to require that the greatest eigenvalue in module satisfies

$$-2 < \lambda_{max}\tau < 0, \tag{B.1}$$

to ensure that all eigenvalues of $A$ are included in the stability region of both methods. The greatest eigenvalue in module of matrix $A$ is $\lambda_{max} = \dfrac{-12\nu}{h^2}$. Hence, condition (B.1) becomes

$$\frac{\tau\nu}{h^2} < \frac{1}{6},$$

which is the CFL condition of both methods (3.1) and (4.3) to ensure the stability in one space dimension.

On the other hand, in the advection-diffusion equation, numerical instabilities occur when the Péclet number is large [66, 89]

$$Pe := \frac{||\beta||h}{2\nu}.$$

## B.2. Definition of $B^{*-}_{DG}(\cdot,\cdot)$

In Chapter 3 we defined $B^{-}_{DG}(\cdot,\cdot)$ as

$$\begin{aligned} B^{-}_{DG}(u,v) := & \sum_{k=1}^{m} \int_{I_k} \Big( \langle u_{,t}, v\rangle + (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v) \Big) dt \\ & + \sum_{k=1}^{m} \left( [\![u]\!]^k, v(t_k^-) \right) + (u(0^+), v(0^-)), \end{aligned} \tag{B.2}$$

if we integrate by parts in time the first integral of (B.2) we obtain

$$\sum_{k=1}^{m} \int_{I_k} -\langle u, v_{,t}\rangle \, dt + \sum_{k=1}^{m} \Big( (u(t_k^-), v(t_k^-)) - (u(t_{k-1}^+), v(t_{k-1}^+)) \Big). \tag{B.3}$$

Now, we add to expression (B.3) the jump terms and the initial condition of (B.2) and we have

$$\sum_{k=1}^{m}\int_{I_k} -\langle u, v_{,t}\rangle\, dt + \sum_{k=1}^{m}(u(t_k^-), v(t_k^-)) - \sum_{k=1}^{m}(u(t_{k-1}^+), v(t_{k-1}^+))$$

$$+\sum_{k=1}^{m}(u(t_k^+), v(t_k^-)) - \sum_{k=1}^{m}(u(t_k^-), v(t_k^-)) + (u(0^+), v(0^-)).$$

Simplifying the last expression

$$\sum_{k=1}^{m}\int_{I_k} -\langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m-1}(u(t_{k-1}^+), v(t_{k-1}^+)) + \sum_{k=0}^{m}(u(t_k^+), v(t_k^-)),$$

Equivalently

$$\sum_{k=1}^{m}\int_{I_k} -\langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m}(u(t_{k-1}^+), v(t_{k-1}^+)) + \sum_{k=0}^{m-1}(u(t_k^+), v(t_k^-)) + (u(T^+), v(T^-)),$$

and fixing the indices, we obtain

$$\sum_{k=1}^{m}\int_{I_k} -\langle u, v_{,t}\rangle\, dt - \sum_{k=1}^{m}(u(t_{k-1}^+), v(t_{k-1}^+)) + \sum_{k=1}^{m}(u(t_{k-1}^+), v(t_{k-1}^-)) + (u(T^+), v(T^-)).$$

Therefore, we have the following expression

$$\sum_{k=1}^{m}\int_{I_k}\Big(-\langle u, v_{,t}\rangle + (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v)\Big)dt - \sum_{k=1}^{m}\big(u(t_{k-1}^+), [\![v]\!]^{k-1}\big) + (u(T^+), v(T^-)).$$

Finally, integrating by parts in space the advection term and as the advection field is divergence free ($\nabla\cdot\beta = 0$), we obtain

$$\sum_{k=1}^{m}\int_{I_k}\Big(-\langle u, v_{,t}\rangle + (\nabla u, \nu\nabla v) - (u, \beta\cdot\nabla v)\Big)dt - \sum_{k=1}^{m}\big(u(t_{k-1}^+), [\![v]\!]^{k-1}\big) + (u(T^+), v(T^-)).$$

Then, we define the discontinuous Galerkin bilinear form for the dual problem as

$$B^{*-}_{DG}(z, v) := \sum_{k=1}^{m}\int_{I_k}\Big(-\langle z_{,t}, v\rangle + (\nu\nabla z, \nabla v) - (\beta\cdot\nabla z, v)\Big)dt$$
$$-\sum_{k=1}^{m}\big([\![z]\!]^{k-1}, v(t_{k-1}^+)\big) + (z(T^-), v(T^+)).$$

# C. $LU$ factorization of Kronecker product matrices and residual minimization methods

In this appendix, we explain how to perform the $LU$ factorization of a system of linear equations where the matrix of the system can be expressed as Kronecker product of matrices and solve it in linear computational cost. We also derive the minimization problem employed in Section 7.2 from a linear weak problem.

## C.1. $LU$ factorization

In this section, we explain how authors in [99, 100] perform the $LU$ factorization of the following system of linear equations

$$\mathbf{M}\mathbf{x} = \mathbf{c}, \tag{C.1}$$

where $\mathbf{M} = \mathbf{A} \otimes \mathbf{B}$, $\mathbf{A}$ is an $n \times n$ invertible matrix and $\mathbf{B}$ is $m \times m$. From the definition of Kronecker product of matrices, we have

$$\mathbf{M} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}B_{11} & \mathbf{A}B_{12} & \cdots & \mathbf{A}B_{1m} \\ \mathbf{A}B_{21} & \mathbf{A}B_{22} & \cdots & \mathbf{A}B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}B_{m1} & \mathbf{A}B_{m2} & \cdots & \mathbf{A}B_{mm} \end{bmatrix}.$$

We partition the right-hand-side and the solution vectors into $m$ blocks of size $n$

$$\mathbf{x}_i = \left[x_{i1} \ \dots \ x_{in}\right]^T,$$
$$\mathbf{c}_i = \left[c_{i1} \ \dots \ c_{in}\right]^T,$$

for $i = 1, ..., m$. We can rewrite system (C.1) as follows

$$\begin{cases} \mathbf{A}B_{11}\mathbf{x}_1 + \mathbf{A}B_{12}\mathbf{x}_2 + \cdots + \mathbf{A}B_{1m}\mathbf{x}_m = \mathbf{c}_1, \\ \mathbf{A}B_{21}\mathbf{x}_1 + \mathbf{A}B_{22}\mathbf{x}_2 + \cdots + \mathbf{A}B_{2m}\mathbf{x}_m = \mathbf{c}_2, \\ \qquad\vdots \qquad\qquad\quad \vdots \quad\ \vdots \qquad\qquad\ \vdots \\ \mathbf{A}B_{m1}\mathbf{x}_1 + \mathbf{A}B_{m2}\mathbf{x}_2 + \cdots + \mathbf{A}B_{mm}\mathbf{x}_m = \mathbf{c}_m, \end{cases}$$

and equivalently

$$\begin{cases} \mathbf{A}\big(B_{11}\mathbf{x}_1 + B_{12}\mathbf{x}_2 \ + \cdots + B_{1m}\mathbf{x}_m\big) \ = \mathbf{c}_1, \\ \mathbf{A}\big(B_{21}\mathbf{x}_1 + B_{22}\mathbf{x}_2 \ + \cdots + B_{2m}\mathbf{x}_m\big) \ = \mathbf{c}_2, \\ \qquad\quad \vdots \qquad\quad \vdots \quad\ \vdots \qquad\quad \vdots \\ \mathbf{A}\big(B_{m1}\mathbf{x}_1 + B_{m2}\mathbf{x}_2 + \cdots + B_{mm}\mathbf{x}_m\big) = \mathbf{c}_m. \end{cases} \tag{C.2}$$

We multiply by $\mathbf{A}^{-1}$ in (C.2) and define by $\mathbf{y}_i = \mathbf{A}^{-1}\mathbf{c}_i,\ \forall i = 1, \ldots, m$. Therefore, we have a 1D problem $\mathbf{A}\mathbf{y}_i = \mathbf{c}_i$ with multiple right-hand-sides. Now, we have in (C.2) that

$$\begin{cases} B_{11}\mathbf{x}_1 + B_{12}\mathbf{x}_2 \ + \cdots + B_{1m}\mathbf{x}_m \ = \mathbf{y}_1, \\ B_{21}\mathbf{x}_1 + B_{22}\mathbf{x}_2 \ + \cdots + B_{2m}\mathbf{x}_m \ = \mathbf{y}_2, \\ \qquad\quad \vdots \qquad\quad \vdots \quad\ \vdots \qquad\quad \vdots \\ B_{m1}\mathbf{x}_1 + B_{m2}\mathbf{x}_2 + \cdots + B_{mm}\mathbf{x}_m = \mathbf{y}_m, \end{cases}$$

and we consider each component of $\mathbf{x}_i$ and $\mathbf{y}_i$ to obtain a family of linear systems

$$\begin{cases} B_{11}x_{1j} + B_{12}x_{2j} \ + \cdots + B_{1m}x_{mj} \ = y_{1j}, \\ B_{21}x_{1j} + B_{22}x_{2j} \ + \cdots + B_{2m}x_{mj} \ = y_{2j}, \\ \qquad\quad \vdots \qquad\quad \vdots \quad\ \vdots \qquad\quad \vdots \\ B_{m1}x_{1j} + B_{m2}x_{2j} + \cdots + B_{mm}x_{mj} = y_{mj}, \end{cases}$$

for each $j = 1, \ldots, n$. Here, we have another 1D problem with multiple right-hand-sides $\mathbf{B}\mathbf{x}_i = \mathbf{y}_i$. This method can be recursively extended to Kronecker product matrices with more components, for example, resulting from 3D Finite Element Method (FEM) computations on tensor product meshes with tensor product basis functions.

## C.2. Minimum residual method

In this section, as in [47], we derive the minimum residual method defined in (7.19) from a linear variational problem of the form

$$\begin{cases} \text{Find } u \in U \text{ such that} \\ \ b(u, v) = l(v),\ \forall v \in V. \end{cases} \tag{C.3}$$

Here, the trial $U$ and test $V$ spaces are Hilbert spaces, $b(\cdot, \cdot)$ is a continuous bilinear form on $U \times V$ and $l(\cdot)$ is a continuous linear form in $V$. For the weak problem (C.3), we define an operator $B : U \to V'$ such that

$$\langle Bu, v\rangle_{V' \times V} = b\,(u, v)\,,$$

being $V'$ the dual space of $V$ so we can reformulate the problem (C.3) as

$$Bu - l = 0.$$

Let $U_h \subset U$ be a discrete subspace of $U$. We wish to minimize the residual

$$u_h = \underset{w_h \in U_h}{\arg\min} \frac{1}{2} ||Bw_h - l||^2_{V'}. \tag{C.4}$$

Now, we introduce the Riesz operator which an isometric isomorphism

$$R_V : V \ni v \rightarrow (v, .) \in V',$$

and we can project problem (C.4) back to $V$ as

$$u_h = \underset{w_h \in U_h}{\arg\min} \frac{1}{2} ||R_V^{-1}(Bw_h - l)||^2_{V}, \tag{C.5}$$

The minimum in (C.5) is attained at $u_h$ when the Gâteaux derivative is equal to zero in all directions

$$\left(R_V^{-1}(Bu_h - l), R_V^{-1}(Bw_h)\right)_V = 0, \ \forall w_h \in U_h. \tag{C.6}$$

Now, we define the error representation function $r = R_V^{-1}(Bu_h - l)$, which can be defined as the solution of the following variational problem

$$\begin{cases} \text{Find } r \in V \text{ such that} \\ (r, v)_V = b(u_h, v) - l(v), \ \forall v \in V. \end{cases} \tag{C.7}$$

We can also rewrite problem (C.6) as

$$\left(r, R_V^{-1}(Bw_h)\right)_V = 0, \ \forall w_h \in U_h,$$

which is equivalent to

$$b(w_h, r) = 0, \ \forall w_h \in U_h. \tag{C.8}$$

Finally, combining (C.7) and (C.8) we obtain the saddle-point problem we employ in Section 7.2

$$\begin{cases} \text{Find } \{r, u_h\} \in V \times U_h \text{ such that} \\ (r, v)_V - b(u_h, v) = -l(v), \ \forall v \in V, \\ \qquad\qquad b(w_h, r) = 0, \ \forall w_h \in U_h. \end{cases}$$

# Bibliography


[1] R. Abedi, B. Petracovici, and R. B. Haber. A space-time discontinuous Galerkin method for linearized elastodynamics with element-wise momentum balance. *Computer Methods in Applied Mechanics and Engineering*, 195(25):3247–3273, 2006. (cited in page(s) 2)

[2] N. Ahmed and V. John. Adaptive time step control for higher order variational time discretizations applied to convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:83–101, 2015. (cited in page(s) 4)

[3] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*, volume 37. John Wiley & Sons, 2011. (cited in page(s) 2, 38)

[4] R. C. Almeida and J. T. Oden. Solution verification, goal-oriented adaptive methods for stochastic advection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2472–2486, 2010. (cited in page(s) 1)

[5] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001. (cited in page(s) 107)

[6] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4):501–520, 2000. (cited in page(s) 107)

[7] A. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Mathematics of Computation*, 52(186):255–274, 1989. (cited in page(s) vii, 2)

[8] L. Bales and I. Lasiecka. Continuous finite elements in space and time for the nonhomogeneous wave equation. *Computers & Mathematics with Applications*, 27(3):91–102, 1994. (cited in page(s) 3)

[9] W. Bangerth, M. Geiger, and R. Rannacher. Adaptive Galerkin finite element methods for the wave equation. *Computational Methods in Applied Mathematics*, 10(1):3–48, 2010. (cited in page(s) vii, 3, 4, 11, 38, 40, 78, 79)

[10] W. Bangerth and R. Rannacher. Finite element approximation of the acoustic wave equation: error control and mesh adaptation. *East West Journal of Numerical Mathematics*, (4):263–282. (cited in page(s) 3)

[11] W. Bangerth and R. Rannacher. Adaptive finite element techniques for the acoustic wave equation. *Journal of Computational Acoustics*, 9(2):575–591, 2001. (cited in page(s) 3, 65)

[12] W. Bangerth and R. Rannacher. *Adaptive finite element methods for differential equations*. Birkhäuser, 2013. (cited in page(s) 3)

[13] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4-5):310–322, 2006. (cited in page(s) 6)

[14] R. Becker, D. Meidner, and B. Vexler. Efficient numerical solution of parabolic optimization problems by finite element methods. *Optimisation Methods and Software*, 22(5):813–833, 2007. (cited in page(s) 38)

[15] R. Becker and R. Rannacher. *A feed-back approach to error control in finite element methods: basic analysis and examples*. Citeseer, 1996. (cited in page(s) vi, 1)

[16] R. Becker and R. Rannacher. *Weighted a posteriori error control in FE methods*. IWR, 1996. (cited in page(s) 1)

[17] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10:1–102, 2001. (cited in page(s) 1)

[18] C. Bernardi and E. Süli. Time and space adaptivity for the second-order wave equation. *Mathematical Models and Methods in Applied Sciences*, 15(02):199–225, 2005. (cited in page(s) 2)

[19] M. Besier and R. Rannacher. Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow. *International Journal for Numerical Methods in Fluids*, 70(9):1139–1166, 2012. (cited in page(s) 3)

[20] G. Birkhoff, R. S. Varga, and D. Young. Alternating direction implicit methods. In *Advances in computers*, volume 3, pages 189–273. Elsevier, 1962. (cited in page(s) 6)

[21] P. B. Bochev and M. D. Gunzburger. *Least-squares finite element methods*, volume 166. Springer Science & Business Media, 2009. (cited in page(s) 5)

[22] C. L. Bottasso. A new look at finite elements in time: a variational interpretation of Runge-Kutta methods. *Applied Numerical Mathematics*, 25(4):355–368, 1997. (cited in page(s) 4)

[23] J. Bramwell, L. Demkowicz, J. Gopalakrishnan, and W. Qiu. A locking-free *hp* DPG method for linear elasticity with symmetric stresses. *Numerische Mathematik*, 122(4):671–707, 2012. (cited in page(s) 5)

[24] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010. (cited in page(s) 2)

[25] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, 1982. (cited in page(s) viii, 5)

[26] J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2008. (cited in page(s) vii, 2, 31, 33, 35)

[27] V. M. Calo, N. F. Brasher, Y. Bazilevs, and T. J. Hughes. Multiphysics model for blood flow and drug transport with application to patient-specific coronary artery flow. *Computational Mechanics*, 43(1):161–177, 2008. (cited in page(s) 6)

[28] V. Carey, D. Estep, A. Johansson, M. Larson, and S. Tavener. Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions. *SIAM Journal on Scientific Computing*, 32(4):2121–2145, 2010. (cited in page(s) 3, 40)

[29] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan. Breaking spaces and forms for the DPG method and applications including Maxwell equations. *Computers & Mathematics with Applications*, 72(3):494–522, 2016. (cited in page(s) 5)

[30] E. A. Celaya and J. J. Anza. BDF-$\alpha$: A multistep method with numerical damping control. *Universal Journal of Computational Mathematics*, 1(3):96–108, 2013. (cited in page(s) 2)

[31] L. Chamoin and P. Ladevèze. A non-intrusive method for the calculation of strict and efficient bounds of calculated outputs of interest in linear viscoelasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 197(9):994–1014, 2008. (cited in page(s) 1)

[32] J. Chan, J. A. Evans, and W. Qiu. A dual Petrov-Galerkin finite element method for the convection-diffusion equation. *Computers & Mathematics with Applications*, 68(11):1513–1529, 2014. (cited in page(s) 5)

[33] J. H. Chaudhry, D. Estep, V. Ginting, J. N. Shadid, and S. Tavener. A posteriori error analysis of IMEX multi-step time integration methods for advection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:730–751, 2015. (cited in page(s) 3, 4)

[34] Z. Chen and J. Feng. An adaptive finite element algorithm with reliable and efficient error control for linear parabolic problems. *Mathematics of Computation*, 73(247):1167–1193, 2004. (cited in page(s) 2)

[35] J. Chung and G. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-$\alpha$ method. *Journal of Applied Mechanics*, 60(2):371–375, 1993. (cited in page(s) 2)

[36] A. Cohen, W. Dahmen, and G. Welper. Adaptivity and variational stabilization for convection-diffusion equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(5):1247–1273, 2012. (cited in page(s) 5)

[37] N. Collier, H. Radwan, L. Dalcin, and V. M. Calo. Time adaptivity in the diffusive wave approximation to the shallow water equations. *Journal of Computational Science*, 4(3):152–156, 2013. (cited in page(s) 2, 3)

[38] J. Collins, D. Estep, and S. Tavener. A posteriori error analysis for finite element methods with projection operators as applied to explicit time integration techniques. *BIT Numerical Mathematics*, 55(4):1017–1042, 2015. (cited in page(s) viii, 5)

[39] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009. (cited in page(s) viii, 6, 98)

[40] V. Darrigrand, D. Pardo, and I. Muga. Goal-oriented adaptivity using unconventional error representations for the 1D Helmholtz equation. *Computers & Mathematics with Applications*, 69(9):964–979, 2015. (cited in page(s) vi, ix, x, 2, 7, 8, 63, 77, 82)

[41] V. Darrigrand, Á. Rodríguez-Rozas, I. Muga, D. Pardo, A. Romkes, and S. Prudhomme. Goal-oriented adaptivity using unconventional error representations for the multidimensional Helmholtz equation. *International Journal for Numerical Methods in Engineering*, 113(1):22–42, 2018. (cited in page(s) vi, ix, x, 2, 7, 8, 63)

[42] M. Delfour and F. Dubeau. Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Mathematics of Computation*, 47(175):169–189, 1986. (cited in page(s) 4)

[43] M. Delfour, W. Hager, and F. Trochu. Discontinuous Galerkin methods for ordinary differential equations. *Mathematics of Computation*, 36(154):455–473, 1981. (cited in page(s) 4)

[44] L. Demkowicz. *Computing with hp-adaptive finite elements. Volume 1. One and two dimensional elliptic and Maxwell problems.* CRC Press, 2006. (cited in page(s) vi, 1)

[45] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. Part I: The transport equation. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1558–1572, 2010. (cited in page(s) viii, 5)

[46] L. Demkowicz and J. Gopalakrishnan. A class of discontinuous Petrov-Galerkin methods. Part II: Optimal test functions. *Numerical Methods for Partial Differential Equations*, 27(1):70–105, 2011. (cited in page(s) viii, 5)

[47] L. Demkowicz and J. Gopalakrishnan. An overview of the discontinuous Petrov Galerkin method. In *Recent developments in discontinuous Galerkin finite element methods for partial differential equations*, pages 149–180. Springer, 2014. (cited in page(s) 5, 103, 113, 127)

[48] L. Demkowicz, J. Gopalakrishnan, and B. Keith. The DPG-star method. *arXiv preprint arXiv:1809.03153*, 2018. (cited in page(s) viii, 5, 111, 114)

[49] L. Demkowicz, J. Gopalakrishnan, S. Nagaraj, and P. Sepúlveda. A space-time dpg method for the Schrödinger equation. *SIAM Journal on Numerical Analysis*, 55(4):1740–1759, 2017. (cited in page(s) 5)

[50] L. Demkowicz, J. Gopalakrishnan, and A. H. Niemi. A class of discontinuous Petrov-Galerkin methods. Part III: Adaptivity. *Applied Numerical Mathematics*, 62(4):396–427, 2012. (cited in page(s) 5)

[51] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszyński, W. Rachowicz, and A. Zdunek. *Computing with hp-adaptive finite elements. Volume 2. Frontiers: three dimensional elliptic and Maxwell problems with applications*. CRC Press, 2007. (cited in page(s) vi, 1)

[52] L. Demkowicz, J. Oden, and W. Rachowicz. A new finite element method for solving compressible Navier-Stokes equations based on an operator splitting method and hp adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 84(3):275–326, 1990. (cited in page(s) 98)

[53] L. Demkowicz, W. Rachowicz, and P. Devloo. A fully automatic hp-adaptivity. *Journal of Scientific Computing*, 17(1-4):117–142, 2002. (cited in page(s) 1)

[54] P. Díez and G. Calderón. Goal-oriented error estimation for transient parabolic problems. *Computational Mechanics*, 39(5):631–646, 2007. (cited in page(s) vii, 3, 11, 40)

[55] J. Douglas and J. E. Gunn. Two high-order correct difference analogues for the equation of multidimensional heat flow. *Mathematics of Computation*, 17(81):71–80, 1963. (cited in page(s) 8, 101)

[56] J. Douglas and J. E. Gunn. A general formulation of alternating direction methods. *Numerische Mathematik*, 6(1):428–453, 1964. (cited in page(s) 8, 101)

[57] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956. (cited in page(s) 100)

[58] J. Erickson, D. Guoy, J. M. Sullivan, and A. Üngör. Building space-time meshes over arbitrary spatial domains. *Engineering with Computers*, 20(4):342–353, 2005. (cited in page(s) 2)

[59] K. Eriksson and C. Johnson. Error estimates and automatic time step control for nonlinear parabolic problems, I. *SIAM Journal on Numerical Analysis*, 24(1):12–23, 1987. (cited in page(s) 2)

[60] A. Ern and J. L. Guermond. *Theory and practice of finite elements*, volume 159. Springer Science & Business Media, 2013. (cited in page(s) 1)

[61] A. Ern, I. Smears, and M. Vohralík. Guaranteed, locally space-time efficient, and polynomial-degree robust a posteriori error estimates for high-order discretizations of parabolic problems. *SIAM Journal on Numerical Analysis*, 55(6):2811–2834, 2017. (cited in page(s) 4)

[62] D. Estep. A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(1):1–48, 1995. (cited in page(s) 4)

[63] D. Estep and D. French. Global error control for the continuous Galerkin finite element method for ordinary differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 28(7):815–852, 1994. (cited in page(s) 4)

[64] D. Estep and A. Stuart. The dynamical behavior of the discontinuous Galerkin method and related difference schemes. *Mathematics of Computation*, 71(239):1075–1103, 2002. (cited in page(s) 5)

[65] P. W. Fick, E. H. van Brummelen, and K. G. Van der Zee. On the adjoint-consistent formulation of interface conditions in goal-oriented error estimation and adaptivity for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 199(49-52):3369–3385, 2010. (cited in page(s) 1)

[66] L. P. Franca, G. Hauke, and A. Masud. Revisiting stabilized finite element methods for the advective–diffusive equation. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1560–1572, 2006. (cited in page(s) 124)

[67] D. A. French. A space-time finite element method for the wave equation. *Computer Methods in Applied Mechanics and Engineering*, 107(1):145–157, 1993. (cited in page(s) vii, 2)

[68] L. Gao and V. M. Calo. Fast isogeometric solvers for explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 274:19–41, 2014. (cited in page(s) 6)

[69] H. Gómez, V. M. Calo, Y. Bazilevs, and T. J. Hughes. Isogeometric analysis of the Cahn—Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4333–4352, 2008. (cited in page(s) 6, 113)

[70] J. Gopalakrishnan, P. Monk, and P. Sepúlveda. A tent pitching scheme motivated by Friedrichs theory. *Computers & Mathematics with Applications*, 70(5):1114–1135, 2015. (cited in page(s) vii, 2)

[71] J. Gopalakrishnan, I. Muga, and N. Olivares. Dispersive and dissipative errors in the DPG method with scaled norms for Helmholtz equation. *SIAM*

*Journal on Scientific Computing*, 36(1):A20–A39, 2014. (cited in page(s) 5)

[72] J. Gopalakrishnan and P. Sepúlveda. A spacetime DPG method for acoustic waves. *arXiv preprint arXiv:1709.08268*, 2017. (cited in page(s) 5)

[73] D. F. Griffiths and D. J. Higham. *Numerical methods for ordinary differential equations: initial value problems*. Springer Science & Business Media, 2010. (cited in page(s) 123)

[74] J. L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6011–6045, 2006. (cited in page(s) 6)

[75] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006. (cited in page(s) 29, 30)

[76] E. Hairer, S. Nørsett, and G. Wanner. Solving ordinary differential equations I. nonstiff problems, 1987. (cited in page(s) vii, 2)

[77] H. M. Hilber, T. J. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292, 1977. (cited in page(s) 2)

[78] S. S. Hossain, S. F. Hossainy, Y. Bazilevs, V. M. Calo, and T. J. Hughes. Mathematical modeling of coupled drug and drug-encapsulated nanoparticle transport in patient-specific coronary artery walls. *Computational Mechanics*, 49(2):213–242, 2012. (cited in page(s) 6)

[79] T. J. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1-4):387–401, 1995. (cited in page(s) viii, 5)

[80] T. J. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Dover Publications, 2012. (cited in page(s) vi, 1, 2)

[81] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005. (cited in page(s) viii, 6, 98, 113)

[82] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):3–24, 1998. (cited in page(s) viii, 5)

[83] T. J. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173–189, 1989. (cited in page(s) 5)

[84] T. J. Hughes and G. M. Hulbert. Space-time finite element methods for elastodynamics: formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66(3):339–363, 1988. (cited in page(s) 2)

[85] G. M. Hulbert and T. J. Hughes. Space-time finite element methods for second-order hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 84(3):327–348, 1990. (cited in page(s) 2)

[86] B. L. Hulme. One-step piecewise polynomial Galerkin methods for initial value problems. *Mathematics of Computation*, 26(118):415–426, 1972. (cited in page(s) 4)

[87] S. Hussain, F. Schieweck, and S. Turek. Higher-order Galerkin time discretizations and fast multigrid solvers for the heat equation. *Journal of Numerical Mathematics*, 19(1):41–61, 2011. (cited in page(s) 4)

[88] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-$\alpha$ method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3):305–319, 2000. (cited in page(s) 2)

[89] V. John and E. Schmeyer. Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion. *Computer Methods in Applied Mechanics and Engineering*, 198(3-4):475–494, 2008. (cited in page(s) 124)

[90] C. Johnson. Discontinuous Galerkin finite element methods for second order hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, 107(1):117–129, 1993. (cited in page(s) 2)

[91] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2012. (cited in page(s) 1)

[92] B. Keith, A. Vaziri Astaneh, and L. Demkowicz. Goal-oriented adaptive mesh refinement for non-symmetric functional settings. *arXiv preprint arXiv:1711.01996*, 2017. (cited in page(s) viii, 5, 111, 114)

[93] U. Köcher and M. Bause. Variational space-time methods for the wave equation. *Journal of Scientific Computing*, 61(2):424–453, 2014. (cited in page(s) 4)

[94] G. Kuru, C. V. Verhoosel, K. G. Van der Zee, and E. H. van Brummelen. Goal-adaptive isogeometric analysis with hierarchical splines. *Computer Methods in Applied Mechanics and Engineering*, 270:270–292, 2014. (cited in page(s) 1)

[95] S. Larsson and V. Thomée. *Partial differential equations with numerical methods*, volume 45. Springer Science & Business Media, 2008. (cited in page(s) 1)

[96] J. L. Lions. *Optimal control of systems governed by partial differential equations*, volume 170. Springer Verlag, 1971. (cited in page(s) 2, 78)

[97] M. Łoś, J. Muñoz-Matute, I. Muga, and M. Paszyński. Isogeometric residual minimization method (iGRM) with direction splitting for implicit problems. *Submitted to Computers & Mathematics with Applications*, December 2018. (cited in page(s) viii, x, xvii, xviii, 6, 8, 107, 108, 109)

[98] M. Łoś, J. Muñoz-Matute, K. Pingali, I. Muga, and M. Paszyński. Parallel shared-memory isogeometric residual minimization (iGRM) simulations of 3D advection-diffusion problems. *Submitted to Engineering with Computers*, April 2019. (cited in page(s) viii, x, xvii, 6, 8, 107, 109, 110, 111)

[99] M. Łoś, M. Paszyński, A. Kłusek, and W. Dzwinel. Application of fast isogeometric L2 projection solver for tumor growth simulations. *Computer Methods in Applied Mechanics and Engineering*, 316:1257–1269, 2017. (cited in page(s) 6, 102, 126)

[100] M. Łoś, M. Woźniak, M. Paszyński, A. Lenharth, M. A. Hassaan, and K. Pingali. IGA-ADS: isogeometric analysis FEM using ADS solver. *Computer Physics Communications*, 217:99–116, 2017. (cited in page(s) 6, 102, 126)

[101] S. T. Miller and R. B. Haber. A spacetime discontinuous Galerkin method for hyperbolic heat conduction. *Computer Methods in Applied Mechanics and Engineering*, 198(2):194–209, 2008. (cited in page(s) 2)

[102] M. Morandi Cecchi and R. Nociforo. Discrete finite elements method in space-time domain for parabolic linear problems. *Le Matematiche*, 46(2):655–673, 1993. (cited in page(s) 3)

[103] I. Muga, M. J. Tyler, and K. G. van der Zee. The discrete-dual minimal-residual method (DDMRes) for weak advection-reaction problems in Banach spaces. *arXiv preprint arXiv:1808.04542*, 2018. (cited in page(s) 5)

[104] I. Muga and K. G. van der Zee. Discretization of linear problems in Banach spaces: residual minimization, nonlinear Petrov-Galerkin, and monotone mixed methods. *arXiv preprint arXiv:1511.04400*, 2015. (cited in page(s) 5)

[105] J. Muñoz-Matute, E. Alberdi, D. Pardo, and V. M. Calo. Time-domain goal-oriented adaptivity using pseudo-dual error representations. *Computer Methods in Applied Mechanics and Engineering*, 325:395–415, 2017. (cited in page(s) viii, x, 6, 8)

[106] J. Muñoz-Matute, V. M. Calo, D. Pardo, E. Alberdi, and K. G. van der Zee. Explicit-in-time goal-oriented adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 347:176–200, 2019. (cited in page(s) viii, ix, 6, 7)

[107] J. Muñoz-Matute, D. Pardo, V. M. Calo, and E. Alberdi. Forward-in-time goal-oriented adaptivity. *International Journal for Numerical Methods in Engineering*, page 1–16, 2019. (cited in page(s) viii, ix, 6, 7)

[108] J. Muñoz-Matute, D. Pardo, V. M. Calo, and E. Alberdi. Variational formulations for explicit Runge-Kutta methods. *Submitted to Finite Elements in Analysis & Design*, December 2018. (cited in page(s) viii, ix, 6, 7)

[109] S. Nicaise and N. Soualem. A posteriori error estimates for a nonconforming finite element discretization of the heat equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(2):319–348, 2005. (cited in page(s) 2)

[110] J. T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & Mathematics with Applications*, 41(5):735–756, 2001. (cited in page(s) vi, 1)

[111] J. T. Oden and S. Prudhomme. Estimation of modeling error in computational mechanics. *Journal of Computational Physics*, 182(2):496–515, 2002. (cited in page(s) 1)

[112] D. Pardo, L. Demkowicz, C. Torres-Verdín, and M. Paszyński. Two-dimensional high-accuracy simulation of resistivity Logging-While-Drilling (LWD) measurements using a self-adaptive goal-oriented *hp* finite element method. *SIAM Journal on Applied Mathematics*, 66(6):2085–2106, 2006. (cited in page(s) 1)

[113] D. Pardo, L. Demkowicz, C. Torres-Verdín, and M. Paszyński. A self-adaptive goal-oriented hp-finite element method with electromagnetic applications. Part II: Electrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 196(37):3585–3597, 2007. (cited in page(s) 1)

[114] D. Pardo, L. Demkowicz, C. Torres-Verdín, and L. Tabarovsky. A goal-oriented hp-adaptive finite element method with electromagnetic applications. Part I: Electrostatics. *International Journal for Numerical Methods in Engineering*, 65(8):1269–1309, 2006. (cited in page(s) 1, 81)

[115] D. Pardo, C. Torres-Verdín, and L. Demkowicz. Simulation of multifrequency borehole resistivity measurements through metal casing using a goal-oriented hp finite-element method. *IEEE transactions on geoscience and remote sensing*, 44(8):2125–2134, 2006. (cited in page(s) 1)

[116] N. Parés, P. Díez, and A. Huerta. Bounds of functional outputs for parabolic problems. Part I: Exact bounds of the discontinuous Galerkin time discretization. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1641–1660, 2008. (cited in page(s) 3, 38, 62)

[117] N. Parés, P. Díez, and A. Huerta. Bounds of functional outputs for parabolic problems. Part II: Bounds of the exact solution. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1661–1679, 2008. (cited in page(s) 3, 62)

[118] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for industrial and Applied Mathematics*, 3(1):28–41, 1955. (cited in page(s) 8, 100)

[119] C. Pozrikidis. *Introduction to finite and spectral element methods using MATLAB*. CRC Press, 2005. (cited in page(s) 12, 23)

[120] S. Prudhomme and J. T. Oden. On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Computer Methods in Applied Mechanics and Engineering*, 176(1):313–331, 1999. (cited in page(s) vi, 1)

[121] S. Prudhomme and J. T. Oden. Computable error estimators and adaptive techniques for fluid flow problems. *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, 25:207–268, 2003. (cited in page(s) 1)

[122] W. Rachowicz, D. Pardo, and L. Demkowicz. Fully automatic hp-adaptivity in three dimensions. *Computer Methods in Applied Mechanics and Engineering*, 195(37-40):4816–4842, 2006. (cited in page(s) 1)

[123] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. Society for Industrial and Applied Mathematics, 2008. (cited in page(s) 13)

[124] A. Romkes, J. T. Oden, and K. Vemaganti. Multi-scale goal-oriented adaptive modeling of random heterogeneous materials. *Mechanics of Materials*, 38(8):859–872, 2006. (cited in page(s) 1)

[125] S. Salsa. *Partial differential equations in action: from modelling to theory*. Springer Science & Business Media, 2008. (cited in page(s) 2, 11)

[126] W. E. Schiesser. *The numerical method of lines: integration of partial differential equations*. Elsevier, 2012. (cited in page(s) vii, 2, 3)

[127] F. Schieweck. A-stable discontinuous Galerkin-Petrov time discretization of higher order. *Journal of Numerical Mathematics*, 18(1):25–57, 2010. (cited in page(s) 2)

[128] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM Journal on Scientific Computing*, 30(1):369–393, 2008. (cited in page(s) 3, 13, 15, 24)

[129] D. Schötzau and C. Schwab. An hp a priori error analysis of the DG time-stepping method for initial value problems. *Calcolo*, 37(4):207–232, 2000. (cited in page(s) 2)

[130] D. Schötzau and C. Schwab. Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method. *SIAM Journal on Numerical Analysis*, 38(3):837–875, 2000. (cited in page(s) vii, 2)

[131] G. Şimşek, X. Wu, K. van der Zee, and E. van Brummelen. Duality-based two-level error estimation for time-dependent PDEs: application to linear and nonlinear parabolic equations. *Computer Methods in Applied Mechanics and Engineering*, 288:83–109, 2015. (cited in page(s) 3, 40, 47, 63)

[132] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. (cited in page(s) ix, 7, 8, 23, 98)

[133] G. Strang and G. J. Fix. *An analysis of the finite element method*, volume 212. Prentice-Hall Englewood Cliffs, NJ, 1973. (cited in page(s) vi, 1)

[134] W. Tang and Y. Sun. Time finite element methods: a unified framework for numerical discretizations of ODEs. *Applied Mathematics and Computation*, 219(4):2158–2179, 2012. (cited in page(s) 5, 19)

[135] J. W. Thomas. *Numerical partial differential equations: finite difference methods*, volume 22. Springer Science & Business Media, 2013. (cited in page(s) 100)

[136] K. G. van der Zee, J. Tinsley Oden, S. Prudhomme, and A. Hawkins-Daarud. Goal-oriented error estimation for Cahn-Hilliard models of binary phase transition. *Numerical Methods for Partial Differential Equations*, 27(1):160–196, 2011. (cited in page(s) vii, 3)

[137] K. G. Van der Zee, E. Van Brummelen, I. Akkerman, and R. De Borst. Goal-oriented error estimation and adaptivity for fluid–structure interaction using exact linearized adjoints. *Computer Methods in Applied Mechanics and Engineering*, 200(37):2738–2757, 2011. (cited in page(s) 1)

[138] K. G. van der Zee, E. Van Brummelen, and R. de Borst. Goal-oriented error estimation and adaptivity for free-boundary problems: The domain-map linearization approach. *SIAM Journal on Scientific Computing*, 32(2):1064–1092, 2010. (cited in page(s) 1)

[139] K. G. van der Zee, E. Van Brummelen, and R. de Borst. Goal-oriented error estimation and adaptivity for free-boundary problems: The shape-linearization approach. *SIAM Journal on Scientific Computing*, 32(2):1093–1118, 2010. (cited in page(s) 1)

[140] K. G. Van der Zee and C. Verhoosel. Isogeometric analysis-based goal-oriented error estimation for free-boundary problems. *Finite Elements in Analysis and Design*, 47(6):600–609, 2011. (cited in page(s) 1)

[141] F. Verdugo, N. Parés, and P. Díez. Error assessment in structural transient dynamics. *Archives of Computational Methods in Engineering*, 21(1):59–90, 2014. (cited in page(s) 3)

[142] F. Verdugo, N. Parés, and P. Díez. Goal-oriented space-time adaptivity for transient dynamics using a modal description of the adjoint solution. *Computational Mechanics*, 54(2):331–352, 2014. (cited in page(s) 3)

[143] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. John Wiley & Sons Inc, 1996. (cited in page(s) 53)

[144] P. Vignal, N. Collier, L. Dalcin, D. Brown, and V. M. Calo. An energy-stable time-integrator for phase-field models. *Computer Methods in Applied Mechanics and Engineering*, 316:1179–1214, 2017. (cited in page(s) 3, 113)

[145] E. L. Wachspress and G. Habetler. An alternating-direction-implicit iteration technique. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):403–423, 1960. (cited in page(s) 6)

[146] G. Wanner and E. Hairer. *Solving ordinary differential equations II*, volume 1. Springer-Verlag, Berlin, 1991. (cited in page(s) 2)

[147] T. Werder, K. Gerdes, D. Schötzau, and C. Schwab. hp-discontinuous Galerkin time stepping for parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 190(49):6685–6708, 2001. (cited in page(s) vii, 2)

[148] W. Wood, M. Bossak, and O. Zienkiewicz. An alpha modification of Newmark's method. *International Journal for Numerical Methods in Engineering*, 15(10):1562–1566, 1980. (cited in page(s) 2)

[149] X. Wu, K. G. Van der Zee, G. Şimşek, and E. H. Van Brummelen. A posteriori error estimation and adaptivity for nonlinear parabolic equations using IMEX-Galerkin discretization of primal and dual equations. *SIAM Journal on Scientific Computing*, 40(5):A3371–A3399, 2018. (cited in page(s) 3, 40)

[150] S. Zhao and G.-W. Wei. A unified discontinuous Galerkin framework for time integration. *Mathematical Methods in the Applied Sciences*, 37(7):1042–1071, 2014. (cited in page(s) viii, 5)

[151] J. Zitelli, I. Muga, L. Demkowicz, J. Gopalakrishnan, D. Pardo, and V. M. Calo. A class of discontinuous Petrov-Galerkin methods. Part IV: The optimal test norm and time-harmonic wave propagation in 1D. *Journal of Computational Physics*, 230(7):2406–2432, 2011. (cited in page(s) 5)